**HOME CREDIT DEFAULT RISK CALCULATION**

**Introduction**

The banks the one which are used to provide loans to the customers. So, home credit is also a banking institute that provides loan. It was the regular issue to check whether the person who seeks for the loan can repay the loan or not. So generally, look at different factors and decide to give the loan or not. The main factor they consider is the credit score of the candidate. So, home credit thought to predict the risk depending on the demographics of the person and the banking history. Here, prediction is a machine learning term. So, we model the existing data and predict the risk of the person. This is a case study taken from Kaggle.

The link that directs to the data: <https://www.kaggle.com/c/home-credit-default-risk/data>

So, the data is given. First, we need to check where the data and also, we need to understand each and every column in all the tables. We need to cleanse the data to the extent possible.

Data description:

There are totally nine tables given. Two of them are application train and application test. Where we find the demographics of the application of the person. The difference between the train and the test is we don’t have a target variable in test we have it in train. The target is the one which is outcome or the one that need to be predicted. If the target is 1 then there is a risk and we need to be careful in giving the loan, if the target is 0 then we don’t have any risk we can give the loan. In nine tables one is tables description where we have details of every tables and every column in the tables. The tables are bureau and bureau balance these are the information about the loan applications from the other banks except home credit. The other table is previous application, this table gives the details about the previous applications in the home credit itself. The other three tables instalment payment, pos cash, sample submission also related to the previous application table. So, this the description of the data given. Now, we need to understand the data and train the model.

The tools used are:

1. postgreSql
2. Anaconda (coding language- python)
3. MS Excel

The postgreSQL is an open-source tool that is used for performing sql operations and we can understand the data clearly by performing multiple quires. The anaconda is used to do python statements, like creating the data frames and understand the data and modelling the data. Which is used for predictions. We used notebook in this. We can also use spyder to scrape the data from the web. Coming to the excel it is also an analytic tool and we use it for sorting and filtering the data. We can look for unique values of the data. We can generate graphs with excel. With python we can obtain the regression graphs to predict.

Things need to be done before prediction:

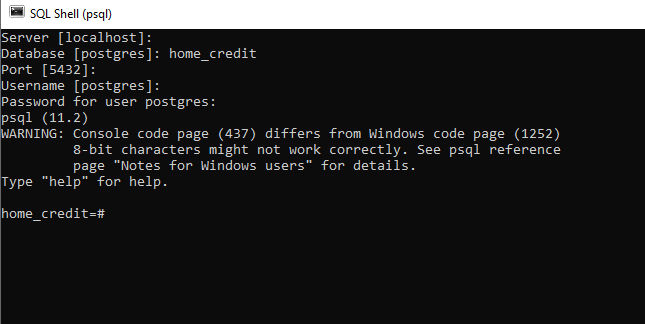
As the data is large we need to look whether we can reduce it. For that we need to find the dependencies, correlations. We also need to check the correlation between each column and the target and between each column. So that if they are highly corelated there is chance of dropping the columns, where in turn we can reduce the data. We also need to check the dependencies by doing the chi square test etc. we also look after what are the derived variables and what are the independent variables.

**Environment setup**

Install postgreSql with latest version.

The link for postgreSql in windows is <https://www.postgresql.org/download/windows/> , its postgreSql version 11. Once it is installed you can use it either from GUI i.e. pgAdmin or from postgreSql shell.

Connecting to postgreSql:



The link for postgreSql for ubuntu use these following commands,

$ sudo apt-get update

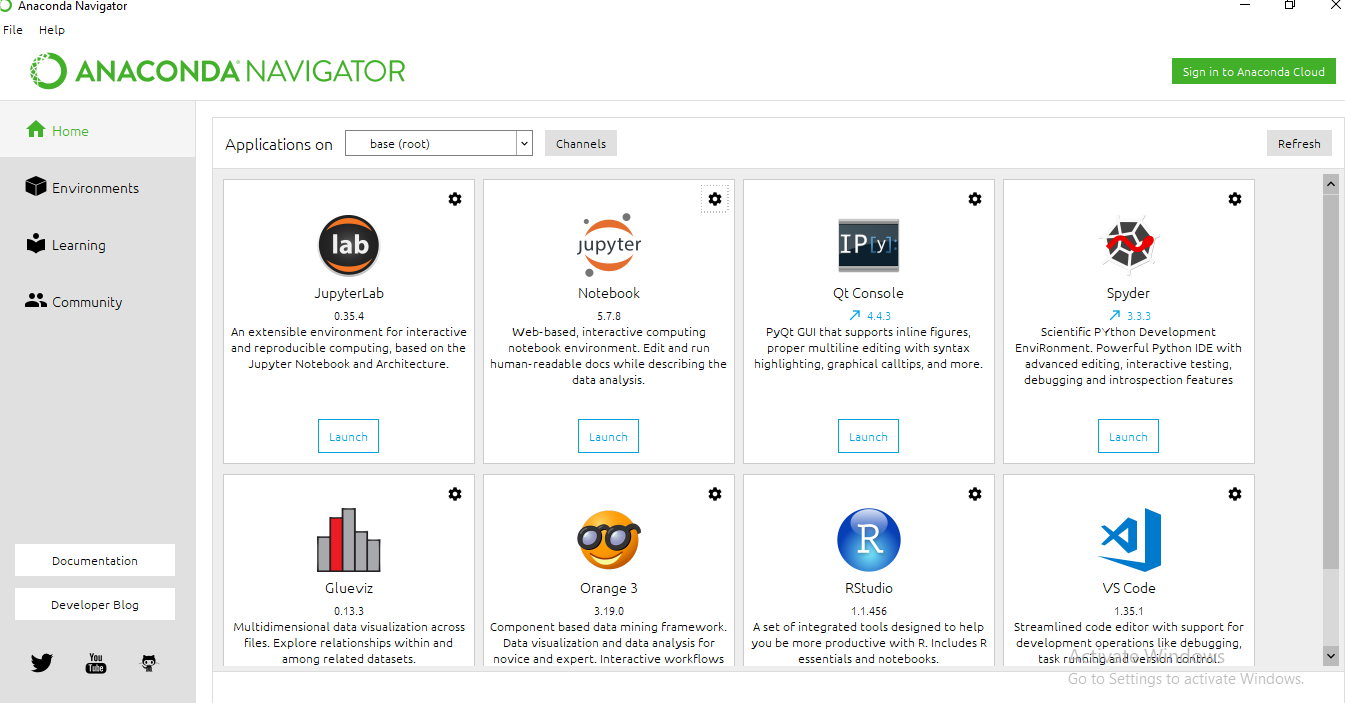
$ sudo apt-get install postgresql postgresql-contrib

Connecting to PostgreSql:

$ sudo su - postgres

$ psql

Install Anaconda latest version for windows, the link for installation is as follows, <https://www.anaconda.com/distribution/> . Once it is installed open the anaconda navigator, where the screen is viewed as follows,



Here we use the Jupyter Notebook where we can write multiple python files, spyder for scrapping. If we want to install or uninstall or update any new packages we to environments and install the requires package and add it to existing python version.

**Understanding the data and reducing the fields**

Observation on application test and train tables:

As columns in the both tables are same except the target, take a copy of application id and target into other table and drop the target column in application train. Later combined both train and test tables to get the large number of rows which in turn helps in better analysis. We need to know what the useful columns in the table. So, look at the correlation between the target and each column and see the pair wise correlations. By performing the pair wise correlations, we can know the dependencies between each column. We can perform correlation we can use python or sql query. Let us look at each table.

Application test and train Tables:

The only difference between application test and train tables is we have target column in train. Irrespective of target totally they are 123 columns. The names of the column were changed for understanding the type of column. The appl\_id acts as a primary key in the table. Where each id refers to a unique loan application. We have target column which states the risk of the application. The columns that we round:

tot\_income

cr\_amt

amt\_annuity

goods\_amt

The following columns apt, barea, expulsion, build, com\_area, elevator, entrance, floor\_max, floor\_min, land\_area, l\_apt, l\_area, nonl\_apt , nonl\_area have mean median and mode. We observed that they are highly correlated. So, take an average of three and consolidate three of each columns into one and by this we can reduce 26 columns.

We have cr\_score1, cr\_score2, cr\_score3 columns are multiplied by 1000.

Reduced 20 documents to 5 docs. The documents that taken into account is doc2, doc4, doc5, doc6, doc7.

m\_phone, e\_phone, w\_phone, phone\_valid phone email is not related to target and most of them submitted these fields.

Bureau table:

select corr(cr\_sum\_amt,cr\_sum\_debt\_amt) from bureau;

0.683419419226711

home\_credit=# select corr(days\_credit,cr\_overdue) from bureau;

corr

---------------------

-0.0272662891381925

(1 row)

home\_credit=# select corr(days\_credit,cr\_max\_overdue) from bureau;

corr

---------------------

-0.0147239106097432

(1 row)

home\_credit=# select corr(cr\_sum\_amt,cr\_sum\_limit\_amt) from bureau;

corr

---------------------

0.00375571598835765

(1 row)

home\_credit=# select corr(cr\_sum\_amt,cr\_sum\_overdue) from bureau;

corr

---------------------

0.00634200375452541

(1 row)

home\_credit=# select corr(cr\_enddate,cr\_ended) from bureau;

corr

-------------------

0.248825102551461

(1 row)

Credit card balance table:

select min(receivable\_amt - receivable\_total\_amt) as diff from ccb1;

diff

--------

-40950

(1 row)

=# select distinct(receivable\_amt, receivable\_total\_amt) from ccb1 where (receivable\_amt - receivable\_total\_amt) = -40950;

row

-----------------

(263632,304582)

(295582,336532)

(304582,345532)

(3 rows)

Highly corealted columns:

dr\_current\_atm

dr\_current

dr\_current\_other

dr\_current\_pos

and

receivable\_principal\_amt

receivable\_amt

receivable\_total\_amt

and

dr\_current\_atm\_cnt

dr\_current\_cnt

dr\_current\_other\_cnt

dr\_current\_pos\_cnt

quires on scores:

home\_credit=# select corr(bin,target) from target t join test\_score a on a.appl\_id = t.appl\_id;

corr

---------------------

-0.0514218837500426

(1 row)

Let do it after multiplying scores with 100 and then do corelations

home\_credit=# select corr(score4,target) from target t join test\_score a on a.appl\_id = t.appl\_id;

corr

---------------------

-0.0618848346675765

(1 row)

home\_credit=# select corr(bin,target) from target t join test\_score a on a.appl\_id = t.appl\_id;

corr

---------------------

-0.0514218837500426

(1 row)

home\_credit=# select bin,count(\*) from test\_score group by bin;

bin | count

-----+-------

| 2

1 | 184

5 | 99

4 | 3027

2 | 3364

6 | 20

3 | 12010

(7 rows)

home\_credit=# select \* from test\_score where bin is null;

appl\_id | score4 | bin

---------+--------+-----

105457 | 500 |

124112 | 500 |

(2 rows)

home\_credit=# select \* from test\_score where score4 = 500;

appl\_id | score4 | bin

---------+--------+-----

105457 | 500 |

124112 | 500 |

(2 rows)

home\_credit=# select corr(bin,score4) from test\_score;

corr

-------------------

0.893177354376497

test\_score t on a.appl\_id = t.appl\_id) to 'C:\Users\ZMHD022\Desktop\home\_credit\TEST\scores.csv' with CSV HEADER;

COPY 18705

User input :

C:\Program Files\PostgreSQL\11\bin>psql -f C:/Users/ZMHD022/Desktop/filesql.sql -h localhost -p 5432 -d home\_credit -U postgres

Password for user postgres:

enter :350761

loan\_type

-----------

CL

(1 row)

C:\Program Files\PostgreSQL\11\bin>

C:\Program Files\PostgreSQL\11\bin>psql -f C:/Users/ZMHD022/Desktop/filesql.sql -h localhost -p 5432 -d home\_credit -U postgres

Password for user postgres:

loan\_type

-----------

CL

(1 row)

C:\Program Files\PostgreSQL\11\bin>psql -f C:/Users/ZMHD022/Desktop/filesql.sql -h localhost -p 5432 -d home\_credit -U postgres

Password for user postgres:

enter :350761

loan\_type

-----------

CL

(1 row)

Application table

home\_credit=# select count(\*) amt\_goods\_price from test1 where amt\_goods\_price is NULL;

amt\_goods\_price

-----------------

278

(1 row)

home\_credit=# select count(\*) amt\_goods\_price from test1 where amt\_goods\_price is NULL and TARGET = 1;

amt\_goods\_price

-----------------

21

(1 row)

home\_credit=# select count(\*) amt\_goods\_price from test1 where amt\_goods\_price is NULL and TARGET = 0;

amt\_goods\_price

-----------------

257

(1 row)

home\_credit=# select count(\*) amt\_goods\_price from test1 where amt\_goods\_price is NULL and amt\_credit is NULL;

amt\_goods\_price

-----------------

0

(1 row)

home\_credit=# select count(\*) amt\_credit from test1 where amt\_credit is NULL;

amt\_credit

------------

0

(1 row)

home\_credit=# select count(\*) target from test1 gruop by name\_education\_type;

ERROR: syntax error at or near "by"

LINE 1: select count(\*) target from test1 gruop by name\_education\_ty...

^

home\_credit=# select count(\*) target from test1 gruopby name\_education\_type;

ERROR: syntax error at or near "name\_education\_type"

LINE 1: select count(\*) target from test1 gruopby name\_education\_typ...

^

home\_credit=# select count(\*) target from test1 group by name\_education\_type;

target

--------

164

74863

10277

3816

218391

(5 rows)

home\_credit=# select count(\*) target, name\_education\_type from test1 group by name\_education\_type;

target | name\_education\_type

--------+-------------------------------

164 | Academic degree

74863 | Higher education

10277 | Incomplete higher

3816 | Lower secondary

218391 | Secondary / secondary special

(5 rows)

home\_credit=# select count(\*) name\_education\_type from test1 group by TARGET having TARGET = 1;

name\_education\_type

---------------------

24825

(1 row)

home\_credit=# select name\_education\_type from test1 group by TARGET having TARGET = 1;

ERROR: column "test1.name\_education\_type" must appear in the GROUP BY clause or be used in an aggregate function

LINE 1: select name\_education\_type from test1 group by TARGET havin...

^

home\_credit=# select count(\*) from test1 group by TARGET having TARGET = 1;

count

-------

24825

(1 row)

home\_credit=# select count(\*) from test1 group by TARGET having TARGET = 0;

count

--------

282686

(1 row)

home\_credit=# select count(\*) target, name\_education\_type from test1 group by name\_education\_type;

target | name\_education\_type

--------+-------------------------------

164 | Academic degree

74863 | Higher education

10277 | Incomplete higher

3816 | Lower secondary

218391 | Secondary / secondary special

(5 rows)

home\_credit=# select count(\*) target, name\_education\_type from test1 group by name\_education\_type where target=1;

ERROR: syntax error at or near "where"

LINE 1: ...ion\_type from test1 group by name\_education\_type where targ...

^

home\_credit=# select count(\*) target, name\_education\_type from test1 group by name\_education\_type,where target=1;

ERROR: syntax error at or near "where"

LINE 1: ...ion\_type from test1 group by name\_education\_type,where targ...

^

home\_credit=# select count(\*) target, name\_education\_type from test1 where target=1 group by name\_education\_type;

target | name\_education\_type

--------+-------------------------------

3 | Academic degree

4009 | Higher education

872 | Incomplete higher

417 | Lower secondary

19524 | Secondary / secondary special

(5 rows)

home\_credit=# select count(\*) target, name\_education\_type from test1 where target=0 group by name\_education\_type;

target | name\_education\_type

--------+-------------------------------

161 | Academic degree

70854 | Higher education

9405 | Incomplete higher

3399 | Lower secondary

198867 | Secondary / secondary special

(5 rows)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit > amt\_goods\_price;

sk\_id\_curr

------------

198763

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit < amt\_goods\_price;

sk\_id\_curr

------------

260

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit = amt\_goods\_price;

sk\_id\_curr

------------

108210

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit = amt\_goods\_price and target =1;

sk\_id\_curr

------------

7305

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit = amt\_goods\_price and target =0;

sk\_id\_curr

------------

100905

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit < amt\_goods\_price and target =1;

sk\_id\_curr

------------

13

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit < amt\_goods\_price and target =0;

sk\_id\_curr

------------

247

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit > amt\_goods\_price target=1;

ERROR: syntax error at or near "target"

LINE 1: ...urr from test1 where amt\_credit > amt\_goods\_price target=1;

^

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit > amt\_goods\_price and target=1;

sk\_id\_curr

------------

17486

(1 row)

home\_credit=# select count(\*) sk\_id\_curr from test1 where amt\_credit > amt\_goods\_price and target=0;

sk\_id\_curr

------------

181277

(1 row)

home\_credit=# SELECT corr(amt\_credit, amt\_goods\_price) from test1;

corr

-------------------

0.986968305423098

(1 row)

home\_credit=# SELECT corr(amt\_credit, amt\_annuity) from test1;

corr

-------------------

0.770138003307034

(1 row)

home\_credit=# SELECT corr(amt\_credit, amt\_income) from test1;

ERROR: column "amt\_income" does not exist

LINE 1: SELECT corr(amt\_credit, amt\_income) from test1;

^

home\_credit=# SELECT corr(amt\_credit, amt\_income\_total) from test1;

corr

-------------------

0.156870271852016

(1 row)

Getting dump of data:

C:\Program Files\PostgreSQL\11\bin>pg\_dump -U postgres -d home\_credit -h 127.0.0.1 > C:/Users/ZMHD022/Downloads/dump1.sql

Password: