Fannie Mae®

An Overview of the Federal National Mortgage Association (FNMA)

- Introduction:
  - The Federal National Mortgage Association (FNMA), commonly known as Fannie Mae, was established in 1938 as a government-sponsored corporation.

- Purpose:
  - Fannie Mae's primary mission is to provide liquidity to the mortgage market.

- Mortgage Loan Acquisition:
  - Fannie Mae purchases mortgage loans from primary lenders, including financial institutions like Bank of America and Wells Fargo.

- Underwriting Standards:
  - Primary lenders follow Fannie Mae's underwriting standards, ensuring that originated loans meet specific criteria set by Fannie Mae for sale.

- Securitization:
  - Fannie Mae sells these acquired mortgages as securities in the bond market.
  - Securitization involves creating pools of mortgages and issuing bonds against them.

- Bond Investors:
  - Bond investors purchase these bonds and receive returns over time through monthly mortgage payments made by mortgage borrowers.

Fannie Mae®

Project Objective: Precision Default Loan Rate Detection

The primary objective of our project is to develop a sophisticated machine learning algorithm that can accurately classify instances of default loan rates within our extensive dataset.

Fannie Mae®

Project Data : Fannie Mae and Economic Data

Home prices are used in mortgage default models because they can provide information about the overall health of the housing market and the potential for default.

In addition to Fannie Mae Dataset we decided to add economical data (unemployment rate in areas)

There are few examples for unemployment data but it's important to ensure that data is reliable and up to date.

We used The Federal Reserve Economic Data (FRED) , and FRED API.

Steps and effects will be explained more clearly in the future.

ECONOMIC RESEARCH
FEDERAL RESERVE BANK of ST. LOUIS

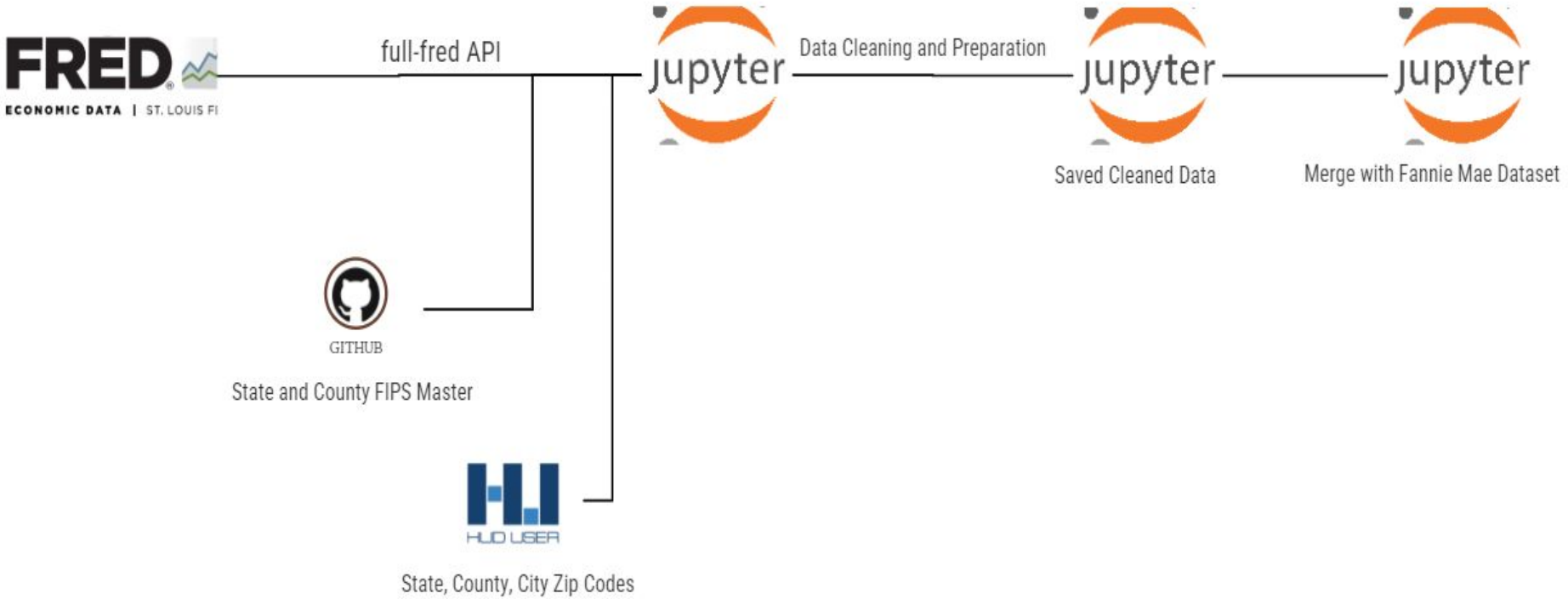
```
from full_fred.fred import Fred

fred = Fred('example.txt')

fred.get_api_key_file()
```

```
'example.txt'
```

```
fred.set_api_key_file('example.txt')
```

```
True
```

Fannie Mae and Economical Data



Fannie Mae®

Project Main Data Overview:

The zipped version of Fannie Mae data is 40GB and becomes close to ~600GB when extracted.

Downloading , Expanding, Sample balancing requires SQL knowledge

In this Phase of Project we used AWS Athena , AWS S3 as a key tool.

aws

Amazon S3

Fannie Mae

Fannie Mae AWS Diagram

Fannie Mae®

Querying Fannie Mae Data With Athena:

For Default Modelling Project we followed some steps in Athena Query.

Our raw data saved in S3 Bucket.

This dataset is pipe-delimited "|".

Sometimes the data types are not read correctly from the csv files. It is recommended to use predetermined data types when reading the file

```
2    `pool_id` string,
3    `loan_id` string,
4    `act_period` string,
5    `channel` string,
6    `seller` string,
7    `servicer` string,
8    `master_servicer` string,
9    `orig_rate` float,
10   `curr_rate` float,
11   `orig_upb` float,
12   `issuance_upb` float,
13   `current_upb` float,
14   `orig_term` float,
15   `orig_date` string,
```

Fannie Mae®

Querying Fannie Mae Data With Athena:

We created two termination states, this is our target variable, by using the performance data and follow the logic:

The Loan is Default if :

‘zero_bal_code’ is equal to one of ['09','03','02','06','15','16']

OR

‘Dlq_status’ is greater than 5

The loan is to be prepaid if:

‘zero_bal_code’ is equal to [01]

Otherwise the loan is still active

```
1  SELECT end_status , count(*) FROM
2  (Select dlq_status ,zero_bal_code,
3  CASE
4  WHEN(zero_bal_code='09')THEN 'D'
5  WHEN(zero_bal_code='03')THEN 'D'
6  WHEN(zero_bal_code='02')THEN 'D'
7  WHEN(zero_bal_code='06')THEN 'D'
8  WHEN(zero_bal_code='15')THEN 'D'
9  WHEN(zero_bal_code='16')THEN 'D'
10 WHEN(zero_bal_code='01')THEN 'P'
11 WHEN (dlq_status not in ('00','01','02','03','04','05')) THEN 'D'
12 WHEN(dlq_status in ('00','01','02','03','04','05') ) THEN 'A'
13 ELSE 'X'
14 END AS end_status
15 from sample_3
16 )
17 GROUP BY end_status
```

Results (4)

Search rows

#	end_status	_col1
1	D	17966047
2	P	34207275
3	A	2293375111
4	X	1

Fannie Mae®

Querying Fannie Mae Data With Athena:

For code development purposes it is recommended to sample down the data by 5-10 % and make code changes. Once you are certain that the code you implemented is correct then you can rerun the code with the original sample size.

We have adjusted the threshold and implemented a more balanced downsampling approach to address the data imbalance issue, primarily stemming from an abundance of active loans compared to default cases."

```
create table "down_sample" as
(select t.*
from(select *,
rand() as random,
CASE
WHEN(zero_bal_code='09')THEN 0.1
WHEN(zero_bal_code='03')THEN 0.1
WHEN(zero_bal_code='02')THEN 0.1
WHEN(zero_bal_code='06')THEN 0.1
WHEN(zero_bal_code='15')THEN 0.1
WHEN(zero_bal_code='16')THEN 0.1
WHEN(zero_bal_code='01')THEN 0.05
WHEN (dlq_status not in ('00','01','02','03','04','05')) THEN 0.1
WHEN(dlq_status in ('00','01','02','03','04','05') ) THEN 0.001
ELSE 0.0001
END AS threshold
from "sample_3") t
where random<=threshold)
```

#	end_status	_col1
1	P	1710115
2	A	2295010
3	D	1796274

Fannie Mae®

Querying Fannie Mae Data With Athena:

Lastly we saved our downsampled data to S3 Bucket for our project.

```
1  CREATE TABLE down_sample_pipe_2
2  WITH ( format = 'TEXTFILE', field_delimiter='|',write_compression = 'NONE',bucketed_by=array['loan_id'],bucket_count=1,
3  external_location = 's3://athenaquarybucket/')
4  AS SELECT *
5  FROM down_sample;
```

Fannie Mae®

Boto3: A Python Library for AWS:

Boto3 is a powerful Python library that allows you to interact with AWS services in an efficient and flexible manner. It provides a simple and intuitive interface for accessing S3 Bucket and Athena, as well as other AWS services. With Boto3, you can easily automate tasks, manage resources, and build applications that leverage the power of AWS.

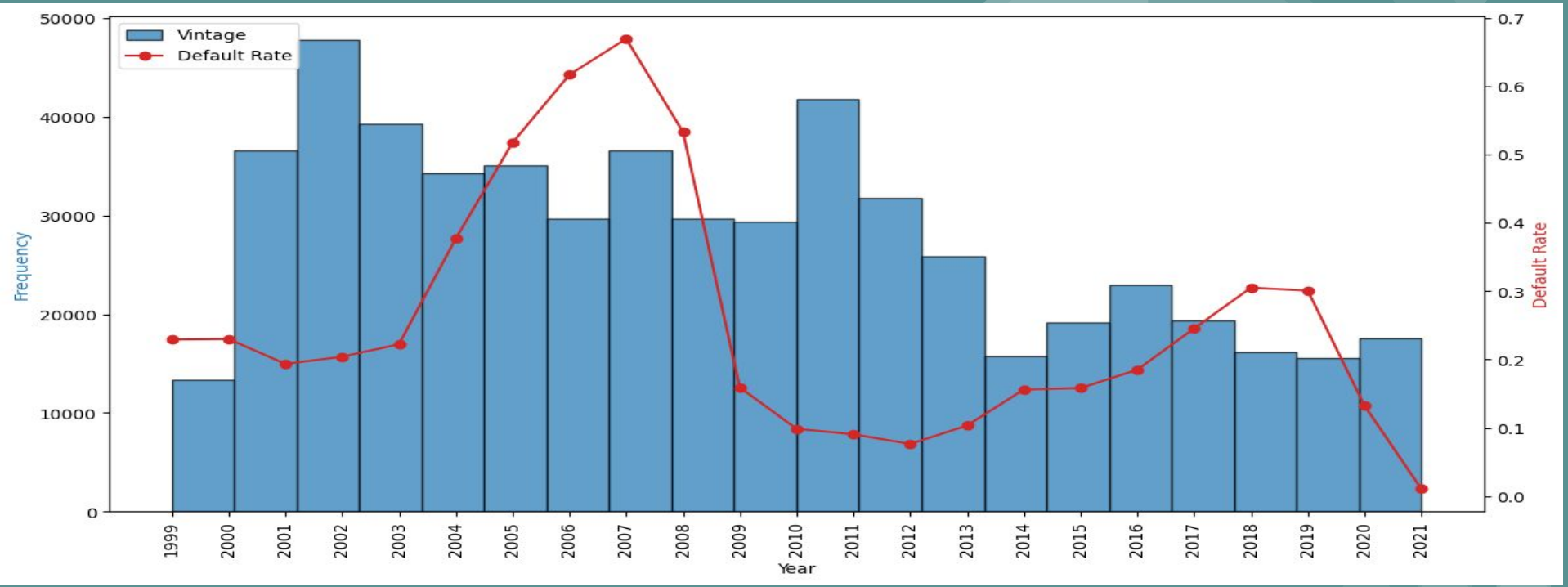
One of the key features of Boto3 is its support for Amazon S3 Bucket. With Boto3, you can easily create, manage, and delete S3 Bucket objects, as well as upload and download files.

```
import boto3
import pandas as pd
```

```
# The S3 bucket where you save your Athena results
S3_BUCKET = 'athenaquarybucket
s3_key="000000_0_bfe06f84-e613-4767-93e2-1d60cb9c214a_20230902_161822_00075_zggzr"
```

```
s3 = boto3.client('s3')
obj = s3.get_object(Bucket=S3_BUCKET, Key=s3_key)
```

Fannie Mae®



Create a variable that captures the origination year (Vintage) of the loan. It is important to see if the model estimation dataset contains enough vintages from different periods of the economic cycles. If you build a model from the vintages originated just before the crisis your model will be very pessimistic, or if you build a model from the vintages in the recent periods the model would be very optimistic

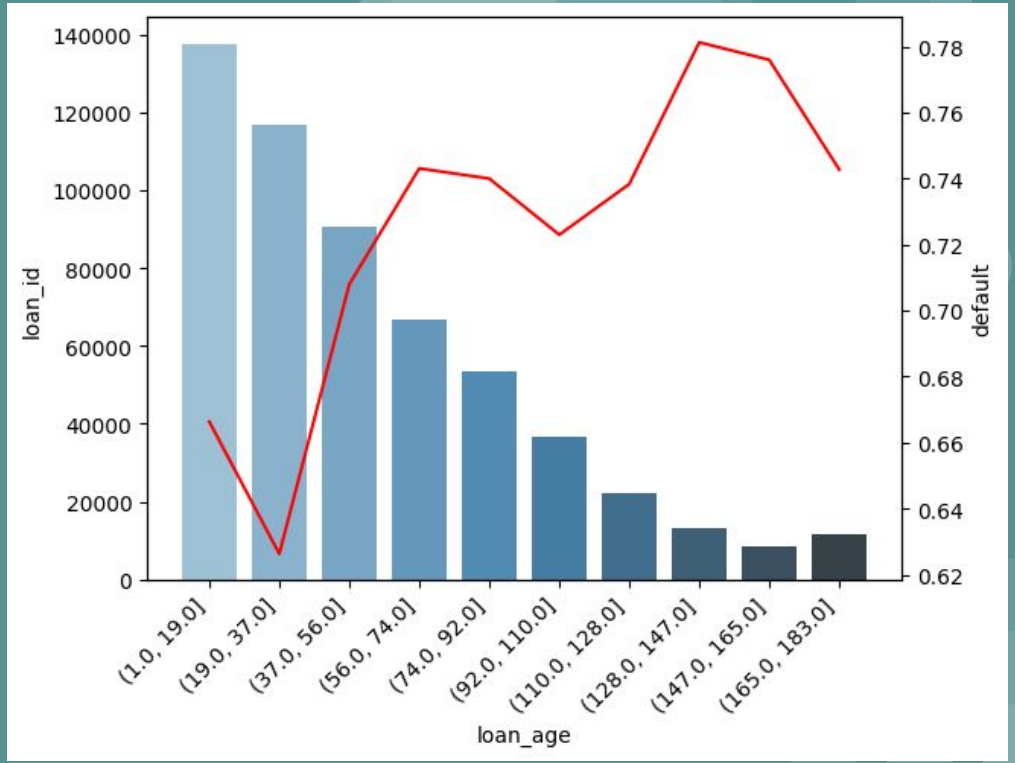
Fannie Mae®

Fannie Mae Data EDA:

The loan_age is calculated as (act_period - orig_date) +1

Loan_age values represents months.

If the length of the term of the loan is longer, the default rate increases. The smaller the term, the less likely borrow to default. 10-13 years old loans have the highest default. (Check Appendixes for code).

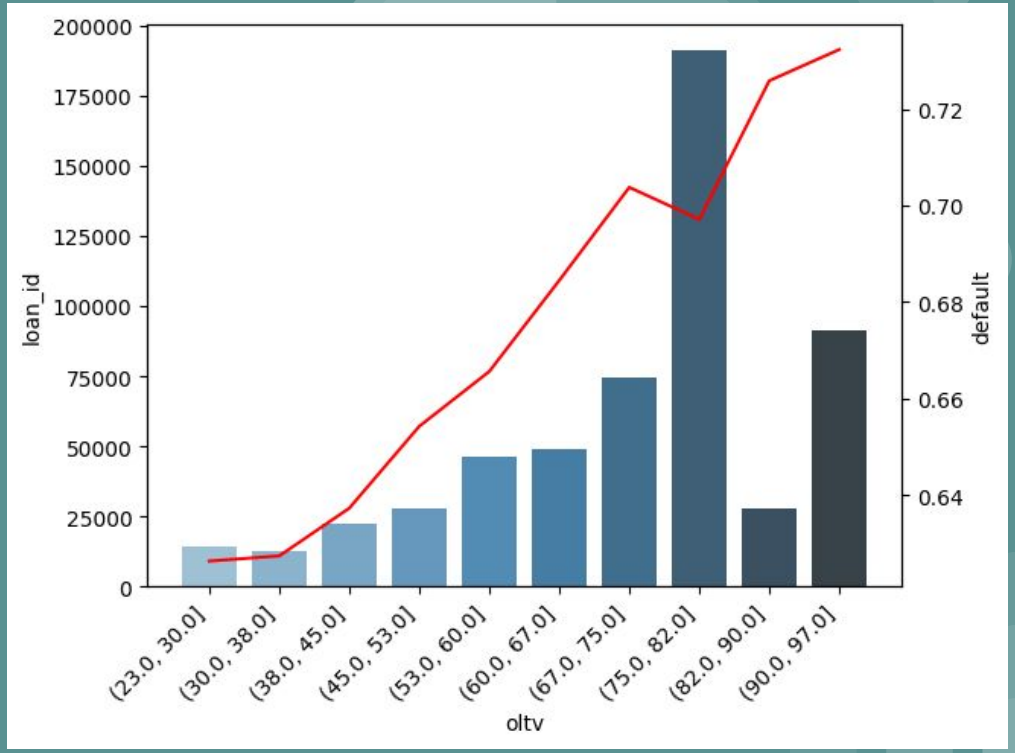


Fannie Mae®

Fannie Mae Data EDA:

The oltv (Original Loan to Value Ratio) is:

The ratio, expressed as a percentage, obtained by dividing the amount of the loan at origination by the value of the property.

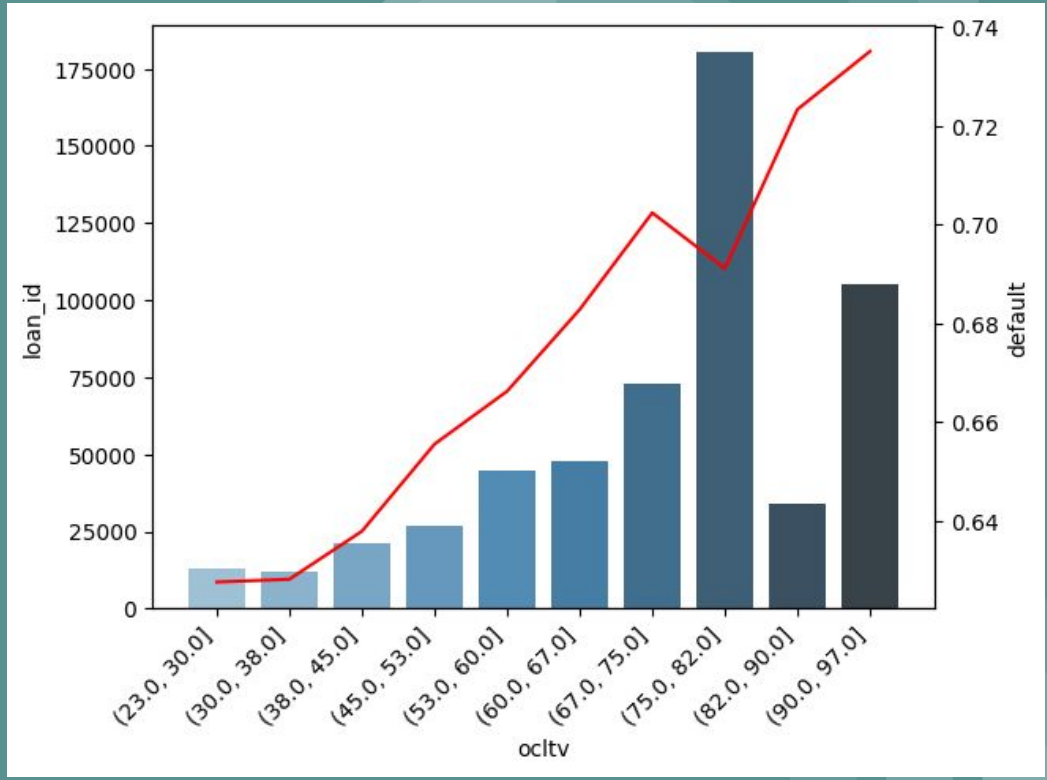


Fannie Mae®

Fannie Mae Data EDA:

The ocltv (Original Combined Loan to Value Ratio is :

The ratio, expressed as a percentage, obtained by dividing the amount of all known outstanding loans at origination by the value of the property.



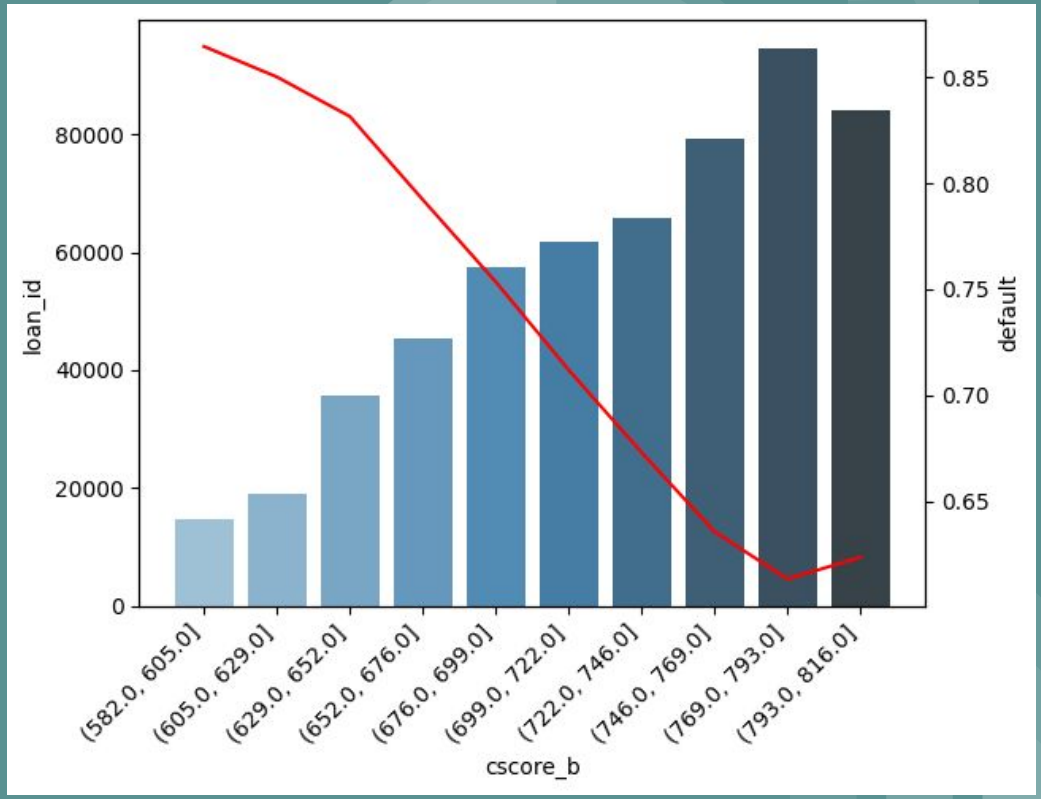
Fannie Mae®

Fannie Mae Data EDA:

The cscore_b (Borrower Credit Score at Origination) is:

A numerical value used by the financial services industry to evaluate the quality of borrower's credit. Credit scores are typically based on a proprietary statistical model that is developed for use by credit data repositories. These credit repositories apply the model to borrower credit information to arrive at a credit score. When this term is used by Fannie Mae, it is referring to the "Classic" FICO score developed by Fair Isaac Corporation.

If the borrower has higher credit score it is less likely to default.

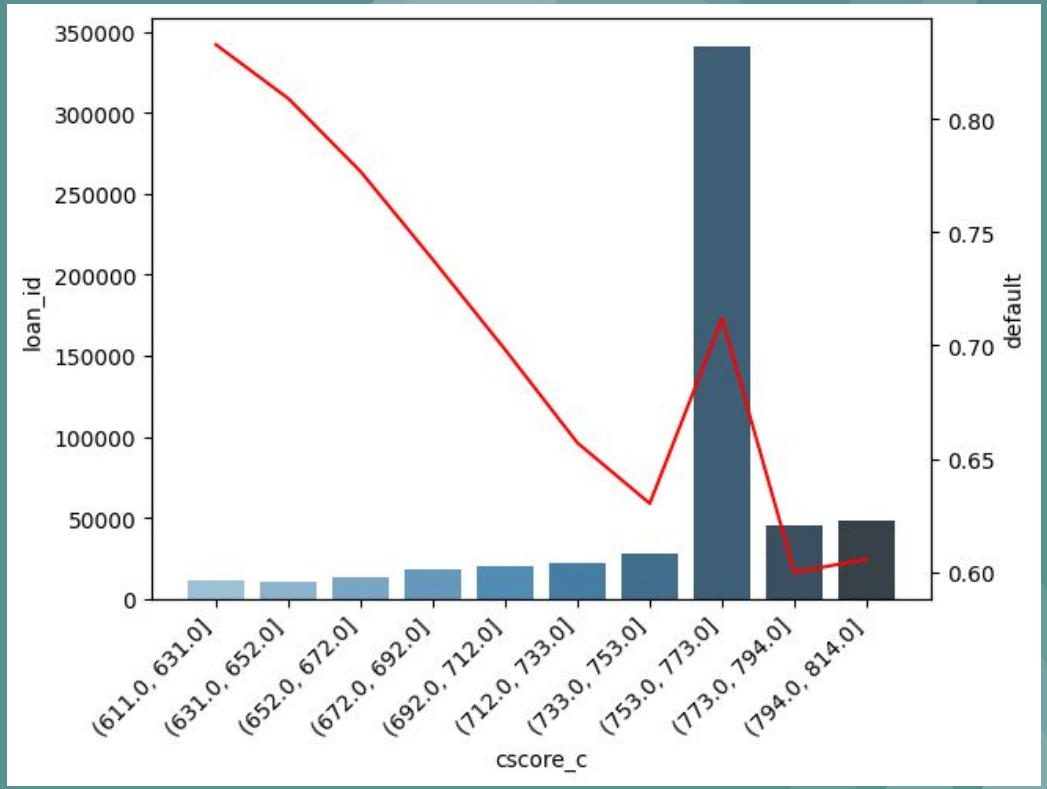


Fannie Mae®

Fannie Mae Data EDA:

The cscore_c (Co-Borrower Credit Score at Origination) is:

A numerical value used by the financial services industry to evaluate the quality of borrower's credit. Credit scores are typically based on a proprietary statistical model that is developed for use by credit data repositories. These credit repositories apply the model to borrower credit information to arrive at a credit score. When this term is used by Fannie Mae, it is referring to the "Classic" FICO score developed by Fair Isaac Corporation.



Fannie Mae®

Fannie Mae Data EDA:

The origination channel used by the party

that delivered the loan to the issuer.

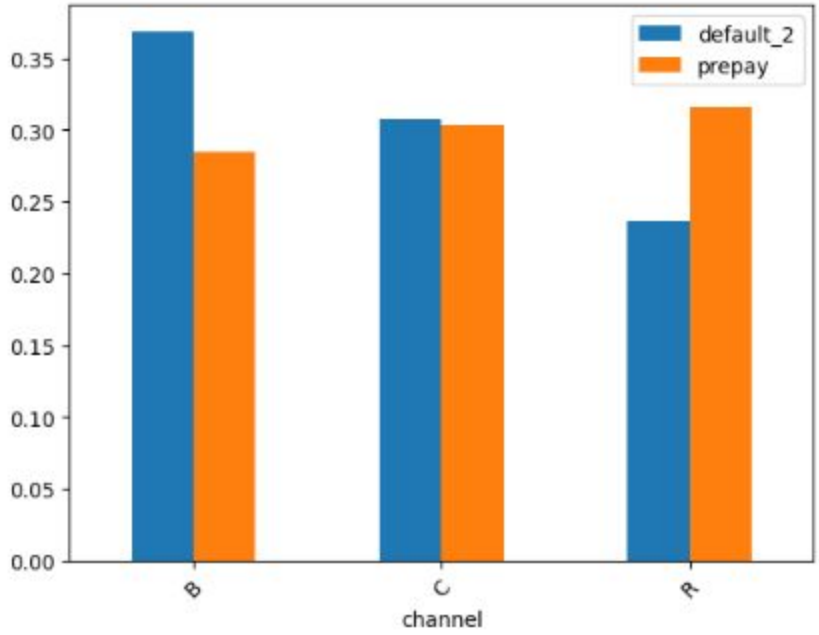
Retail = R;

Correspondent = C;

Broker = B

In a formal context, you can express this statement concisely as follows:
The selection of the 'Broker' mortgage channel is associated with an elevated likelihood of default.

channel	default_2	prepay	channel
B	0.368461	0.284217	B
C	0.307403	0.303709	C
R	0.236647	0.316096	R



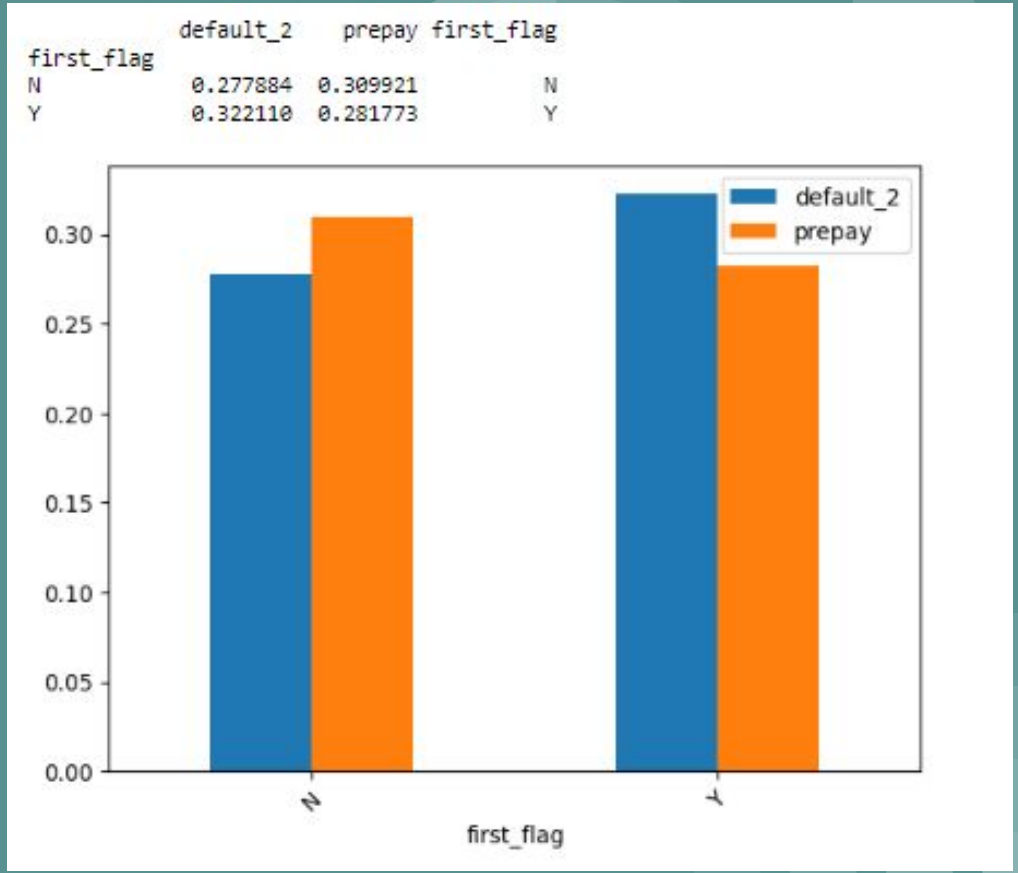
Fannie Mae®

Fannie Mae Data EDA:

An indicator that denotes if the borrower or co-borrower qualifies as a first-time homebuyer.

The outcome of our exploratory data analysis (EDA) indicates that there exists a statistically significant trend, whereby first-time homebuyers exhibit a notably increased likelihood of default when compared to borrowers in subsequent mortgage transactions

first_flag	default_2	prepay	first_flag
N	0.277884	0.309921	N
Y	0.322110	0.281773	Y



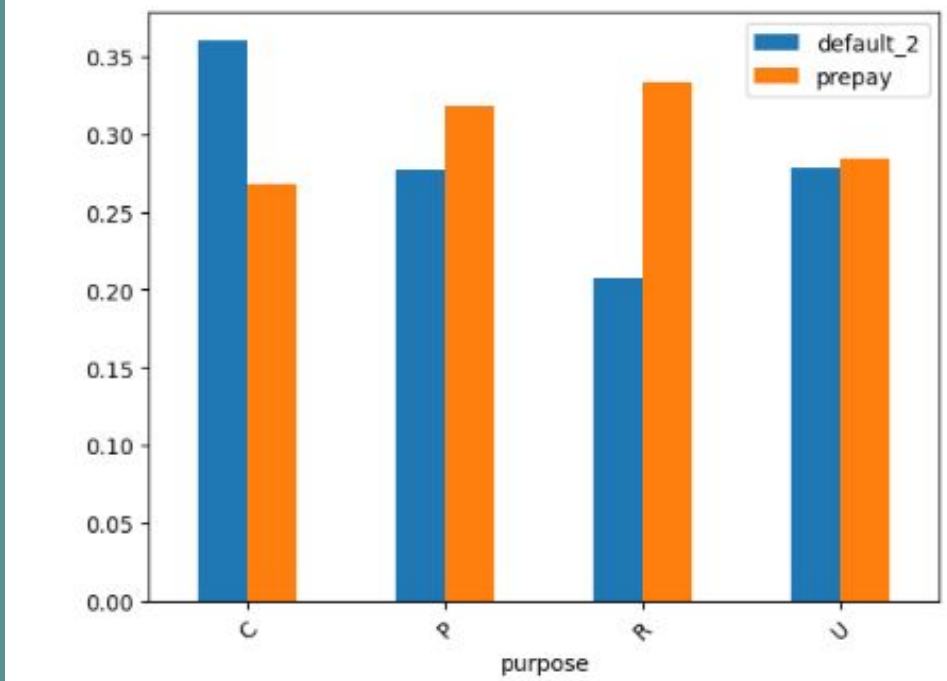
Fannie Mae®

Fannie Mae Data EDA:

An indicator that denotes whether the mortgage loan is either a refinance mortgage or a purchase money mortgage. Purpose may be the purchase of a new property or refinance of an existing lien (with cash out or with no cash out).

Cash-Out Refinance = C
Refinance = R
Purchase = P
Refinance-Not Specified = U

purpose	default_2	prepay	purpose
C	0.360573	0.268541	C
P	0.277741	0.318438	P
R	0.207485	0.333280	R
U	0.278481	0.284810	U



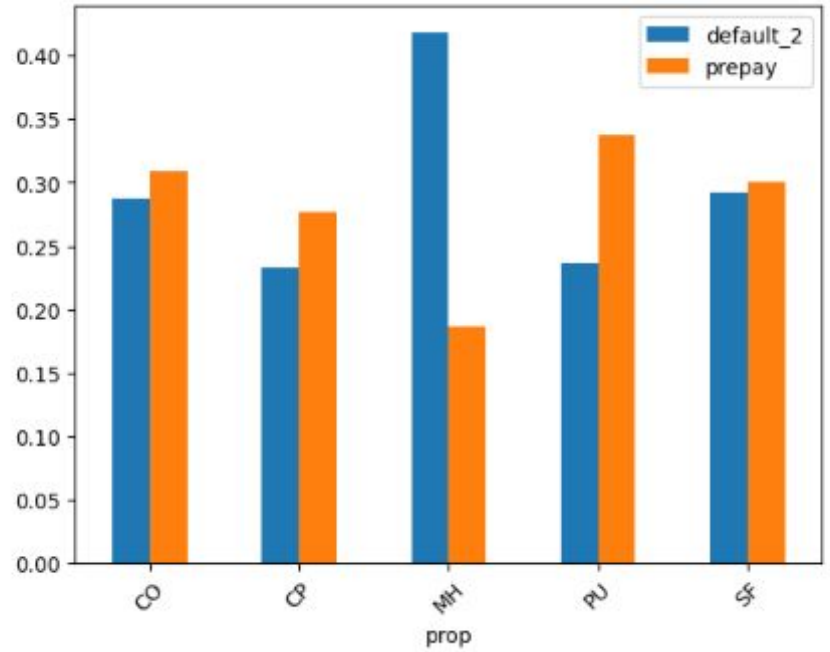
Fannie Mae®

Fannie Mae Data EDA:

If the property is manufactured house there is high likelihood to default.

Condominium = CO
Co-Operative = CP
Planned Urban Development = PU
Manufactured Home = MH
Single-Family Home = SF

prop	default_2	prepay	prop
CO	0.287223	0.309221	CO
CP	0.232403	0.276445	CP
MH	0.418189	0.186310	MH
PU	0.236925	0.337789	PU
SF	0.292189	0.300707	SF



Fannie Mae®

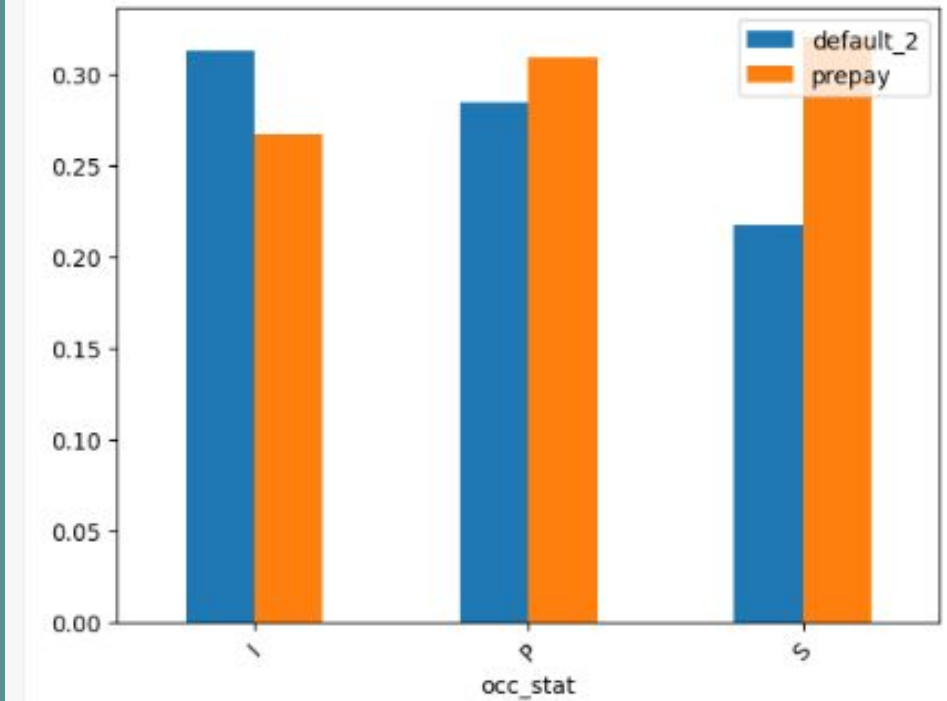
Fannie Mae Data EDA:

The classification describing the property occupancy status at the time the loan was originated.

If the borrower is invested there is a high probability of default.

Principal = P
Second = S
Investor = I

occ_stat	default_2	prepay	occ_stat
I	0.312450	0.267230	I
P	0.283755	0.309129	P
S	0.216493	0.319963	S



Fannie Mae®

APPENDIXES

Fannie Mae®

Exploratory Data Analysis :

Exploratory Data Analysis (EDA) is a critical step performed before modeling, as it provides essential insights into the dataset's characteristics.

We started our EDA with checking missing values.

Checked the percentage of missing values for each column.

Dropped the numerical columns that have missing more than 95%

Fannie Mae uses the same template for different datasets and the empty fields sometimes stand there as placeholders for other reporting purposes. It may be deemed justifiable to exclude these columns from the dataset.

```
obs_count=df_sample.shape[0]
```

```
obs_count
```

```
557564
```

```
missings=(df_sample.isnull().sum()/obs_count).sort_values(ascending=False).to_frame()
```

```
missings
```

	0
first_pay_io	1.00
arm_cap_structure	1.00
arm_index	1.00
months_between_subsequent_payment_reset	1.00
months_until_first_payment_reset	1.00

```
cols_majority_miss=missings.loc[missings>0.95].index.tolist()
```

```
cols_majority_miss
```

```
['first_pay_io',
 'arm_5_yr_indicator',
 'arm_product_type',
 'months_until_first_payment_reset',
 'months_between_subsequent_payment_reset',
 'arm_index',
 'arm_cap_structure',
 'initial_interest_rate_cap',
 'periodic_interest_rate_cap',
 'lifetime_interest_rate_cap',
 'margin',
 'balloon_indicator',
 'plan_number']
```

```
df_sample.drop(cols_majority_miss, inplace=True,axis=1)
```

Fannie Mae®

Exploratory Data Analysis :

For any remaining missing values, we shall employ sample-based imputation techniques as follows:

1. For categorical and indicator variables, we will impute the missing values with the most frequent category.

2. For numeric columns, the missing values will be imputed using either the median or mean, depending on the specific data characteristics and objectives of the analysis.

Creating a 'Vintage' variable is essential to gauge dataset diversity across economic cycles. This ensures that our model isn't overly pessimistic when based on pre-crisis vintages or overly optimistic when reliant on recent vintages.

```
def fill_numerical_values(df,x_var):
    df[x_var] = df[x_var].fillna(df[x_var].median())
```

```
def fill_cat_values(df,x_var):

    # filling with most common class
    df = df.apply(lambda x: x.fillna(x.value_counts().index[0]))
```

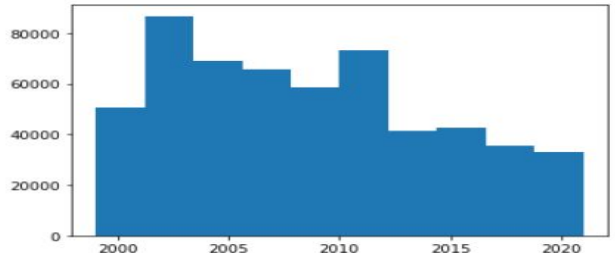
```
num_cols=['orig_rate', 'orig_upb',
       'orig_term' ,'loan_age', 'rem_months',
       'oltv', 'ocltv',  'dti', 'cscore_b', 'cscore_c',
        'mi_pct']
cat_cols=['channel', 'first_flag', 'purpose', 'prop', 'no_units', 'occ_stat', 'state',
       'product', 'ppmt_flg', 'io',
       'mi_type', 'homeready_program_indicator',
       'relocation_mortgage_indicator', 'high_balance_loan_indicator',
       'high_loan_to_value_hltv_refinance_option_indicator','num_bo']
```

```
for i in (num_cols):
  fill_numerical_values(df_sample,i)
```

```
for i in (cat_cols):
  fill_cat_values(df_sample,i)
```

```
df_sample['Vintage']=df_sample['orig_date'].dt.year
plt.hist(df_sample['Vintage'])
```

```
(array([50450., 86797., 69091., 65919., 58771., 73154., 41590., 42583.,
        35713., 33055.]),
 array([1999. , 2001.2, 2003.4, 2005.6, 2007.8, 2010. , 2012.2, 2014.4,
        2016.6, 2018.8, 2021. ]),
 <BarContainer object of 10 artists>)
```



Fannie Mae®

Exploratory Data Analysis :

To assess the relationship with the target variable (default), we will create two separate lists of columns:

1. Continuous Numerical Values

2. Categorical and Indicator Columns

These lists will help us perform a focused analysis to understand how each type of variable relates to the target variable.

We identify outliers in the numerical columns, we will generate box plots. Any extreme addressed by capping and flooring them using the 1st percentile and 99th percentile thresholds specific to each column.

```
def cap_floor(df,x_val):
    q_l = df[x_val].quantile(0.01)
    q_h = df[x_val].quantile(0.99)

    df.loc[df[x_val]<= q_l, x_val] = q_l
    df.loc[df[x_val]>= q_h, x_val] = q_h
```

```
for i in (num_cols):

  cap_floor(df_sample,i)
```

Fannie Mae®

Exploratory Data Analysis :

For each column in the numerical and categorical lists, we will create line plots to visualize the average default rate and bar plots to display the count of observations.

```
def univariate_plot_v3(df,x_var,y_var,bin_size):
   if x_var in num_cols:

       fig,ax=plt.subplots()
       s=df.groupby([pd.cut(df[x_var],bins=bin_size,precision=0)]).agg({'loan_id':'count',y_var:'mean'})
       sns.barplot(x=x_var, y='loan_id', data=s.reset_index(), palette="Blues_d", ax=ax)
       ax2 = ax.twinx()
       sns.lineplot(x=range(len(s.reset_index())), y=y_var, data=s.reset_index(), color='red', markers=True, ax=ax2)
       ax.set_xticklabels(s.index.values, rotation = 45, ha="right")

   elif x_var in cat_cols:
       fig,ax=plt.subplots()
       s=df.groupby(df[x_var]).agg({'loan_id':'count',y_var:'mean'})
       sns.barplot(x=x_var, y='loan_id', data=s.reset_index(), palette="Blues_d", ax=ax)
       ax2 = ax.twinx()
       sns.lineplot(x=range(len(s.reset_index())), y=y_var, data=s.reset_index(), color='red', markers=True, ax=ax2)
       ax.set_xticklabels(s.index.values, rotation = 45, ha="right")
```

```
vars_to_plot=cat_cols+num_cols

for i in (vars_to_plot):

  univariate_plot_v3(df_sample_dflt,i,'Default',10)
```