MICRO CREDIT DEFAULTER PROJECT

Submitted by:

MIENGANDHA SINHA

ACKNOWLEDGMENT

I would like to express my gratitude to the Company FlipRobo-Technology to give this project to give this project to me. In making of this project I hereby used to take help from the references from some websites and also from which is given by the company as sample documentation and details related project and professionals guided me a lot in the project and the other previous projects helped me and guided me in completion of the project.

INTRODUCTION

- Credit Defaulter Problem

  Nowadays, the institution provides small loans to poor clients who typically lack collateral, steady employment, verifiable credit history to improve borrowers from the selection of customers for the credit and the client wants some predictions that could help them in further investment and improvement in selection of customers. The Microfinance Institution provided Group Loans, Agricultural Loans and Individual Business Loans which helps to small community. Micro Credit is the extension of very small loans. It is designed to support entrepreneurship and alleviate poverty. Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

- Conceptual Background of the Domain Problem

  It offers financial services to low income populations and becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. Many major micro finance institutions contribute in this project to uplift those small community. As it is said in the problem-statement that the MFI industry is primarily focusing on low income families and are very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes. So, here the model can be used to predict in terms of a probability for each loan transaction,

whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, label is the target point to predict model, as ‘label 0’ indicates that the loan has not been paid i.e. defaulter or failure and on the other side ‘label 1’ indicates that the loan has been paid i.e. non-defaulter or success. By doing some workout in the project there we can help to predict that how many customers are defaulter or non-defaulter in that area to provide such micro credit and by model training and predicting things makes the previous background to work efficiently.

- Review of Project

  As it is saying the institution from looking at the client database, in order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers. Build a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5days of insurance of loan. From this, there are some variables need to be predict are label (target variable), maxamnt_loans30, maxamnt_loans90, cnt_loans30, loans_loans90. Thus project is all about the financing at the micro level to improve, predict and trained the model of the data given.

- Motivation for the Problem Undertaken

  Here, the datasets having total of 5 rows and 37 columns, no null values, no loan history of some customers, contains numerical and categorical variable, EDA has to be performed to gain relationship of important variable and it label to be either ‘success’ or ‘failure’, to build Machine Learning models, apply regularization, ROC-AUC curve recall-precision, and to determine the optimal values of Hyper parameters. So, to improve the selection of customers for the credit, by predictions of the variable can help to the customers. And for further investment and improvement in selection of customers.

ANALYTICAL PROBLEM FRAMING

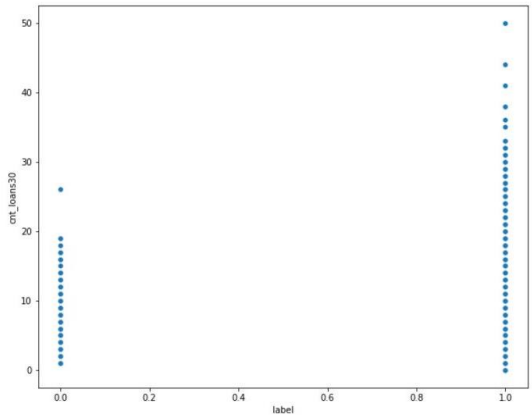
- Mathematical/ Analytical Modelling of the Problem

  In this mathematical/analytical problems are used. They are data frame types, checking the null values by using df.isnull().sum(), the type of data frame is in pandas, data frame info tells that dtypes: float64(18 variables),int64(12 variables), object(2 variables),by using the data visualization the subplot of label,cnt_loans30,cnt_loans90,amnt_loans30,amnt_loans90, and many more variables for distribution plot, line plot, relplot, countplot, scatterplot, boxplot and there are 13 outliers found by counting so , here the formula is used to remove the outliers ahead for further modelling process of variables. From scatterplot here are some target area where doing some work on this project:

1. Number of loans taken by user in last 30 days

In [53]:
```
plt.figure(figsize=(10,8))
sns.scatterplot(df['label'],df['cnt_loans30'])
print(df[['label' , 'cnt_loans30']].corr())
plt.show()
```

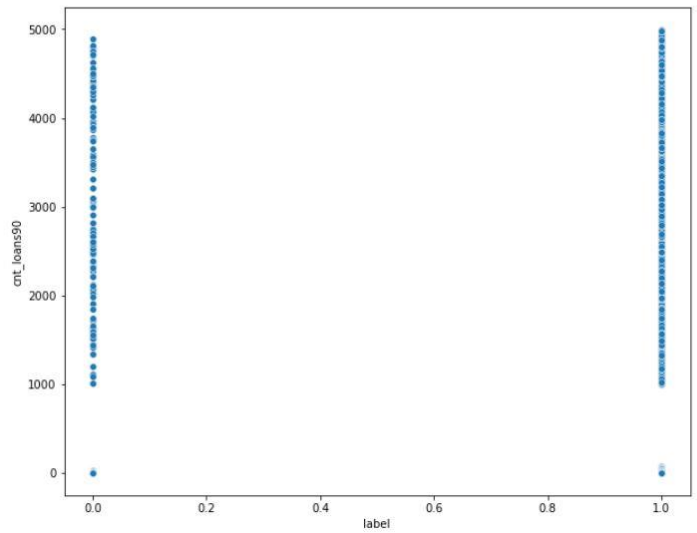
```
              label  cnt_loans30
label       1.000000     0.196283
cnt_loans30 0.196283     1.000000
```



2. Number of loans taken by user in last 90 days

```
plt.figure(figsize=(10,8))
sns.scatterplot(df['label'],df['cnt_loans90'])
print(df[['label' , 'cnt_loans90']].corr())
plt.show()
```

```
              label  cnt_loans90
label       1.000000     0.004733
cnt_loans90 0.004733     1.000000
```



- Data Sources and their formats

  The data sources and their formats are from .csv file (Comma-separated values)

WORKING WITH DATASETS

```
]: 1 df=pd.read_csv(r'C:/Users/user/Desktop/FRTech Internship Project/Micro Credit Project/Data file.csv')
   2 df.head()
```

]:

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...

5 rows × 37 columns

- Data Preprocessing Done

The steps followed for the cleaning of the data is Label Encoder or Encoding after that dropping the attributes of data that target and extra columns show, then from importing preprocessing there transform the target columns into features then lastly it has to set/fit for the data frame.

DATA PREPROCESSING

In [110]:
```
from sklearn import preprocessing
features=df.drop(['cnt_loans90','pcircle','pdate','payback30','payback90'],axis=1)
target=df['cnt_loans90']
col_names=list(features.columns)
scaler=preprocessing.StandardScaler()
features=scaler.fit_transform(features)
features=pd.DataFrame(features,columns=col_names)
features.describe().T
```

Out[110]:

	count	mean	std	min	25%	50%	75%	max
label	209593.0	1.806183e-15	1.000002	-2.647896	0.377658	0.377658	0.377658	0.377658
rental30	209593.0	-3.398164e-16	1.000002	-6.134212	-0.559851	-0.373444	0.154194	45.544863
rental90	209593.0	2.411532e-17	1.000002	-4.887660	-0.551629	-0.372485	0.124494	34.081361
last_rech_date_ma	209593.0	-2.449357e-16	1.000002	-0.070212	-0.069656	-0.069619	-0.069544	18.456181
last_rech_date_da	209593.0	-4.180683e-15	1.000002	-0.070093	-0.069550	-0.069550	-0.069550	18.650400
last_rech_amt_ma	209593.0	6.270109e-16	1.000002	-0.870790	-0.546003	-0.221637	0.103151	22.328322
cnt_ma_rech30	209593.0	1.151419e-15	1.000002	-0.934677	-0.699718	-0.229802	0.240114	46.761802
fr_ma_rech30	209593.0	6.759602e-16	1.000002	-0.069670	-0.069670	-0.069633	-0.069558	18.564580
sumamnt_ma_rech30	209593.0	-1.859425e-16	1.000002	-0.759843	-0.607963	-0.303415	0.227376	79.134453
medianamnt_ma_rech30	209593.0	1.131901e-15	1.000002	-0.875394	-0.503568	-0.132224	0.053689	25.683624
medianmarechprebal30	209593.0	-3.683456e-16	1.000002	-0.075027	-0.071120	-0.070696	-0.069787	18.435414
cnt_ma_rech90	209593.0	1.006487e-15	1.000002	-0.877941	-0.599911	-0.321880	0.234181	45.831198
fr_ma_rech90	209593.0	-1.440573e-16	1.000002	-0.612919	-0.612919	-0.454065	0.022495	6.376633
sumamnt_ma_rech90	209593.0	8.445273e-17	1.000002	-0.735342	-0.597898	-0.306697	0.213776	55.798643
medianamnt_ma_rech90	209593.0	1.331685e-15	1.000002	-0.895719	-0.524383	-0.156410	0.028537	25.525304
medianmarechprebal90	209593.0	2.230007e-16	1.000002	-0.790937	-0.209703	-0.151742	-0.034439	112.033637
cnt_da_rech30	209593.0	-2.297900e-15	1.000002	-0.062759	-0.062759	-0.062759	-0.062759	23.818005
fr_da_rech30	209593.0	3.152777e-15	1.000002	-0.069583	-0.069583	-0.069583	-0.069583	18.484819
cnt_da_rech90	209593.0	-1.478183e-15	1.000002	-0.104375	-0.104375	-0.104375	-0.104375	95.479954
fr_da_rech90	209593.0	7.274570e-15	1.000002	-0.048048	-0.048048	-0.048048	-0.048048	67.222406
cnt_loans30	209593.0	-7.098037e-16	1.000002	-1.080049	-0.688582	-0.297116	0.485818	18.493283
amnt_loans30	209593.0	-3.022465e-16	1.000002	-1.032930	-0.687700	-0.342470	0.347991	16.573818
maxamnt_loans30	209593.0	-2.741964e-16	1.000002	-0.064698	-0.063284	-0.063284	-0.063284	23.459112
medianamnt_loans30	209593.0	-1.705900e-15	1.000002	-0.247794	-0.247794	-0.247794	-0.247794	13.511268
amnt_loans90	209593.0	4.109753e-16	1.000002	-0.893297	-0.666624	-0.439950	0.240070	15.653863
maxamnt_loans90	209593.0	2.875035e-15	1.000002	-3.186113	-0.334212	-0.334212	-0.334212	2.517690
medianamnt_loans90	209593.0	4.533210e-15	1.000002	-0.229594	-0.229594	-0.229594	-0.229594	14.718755

- Data Inputs- Logic- Output Relationships

  The relationships between inputs and outputs can be studied extracting weights of the trained model. Regression is that relationships between them can be blocky or highly structured based on the training data. It requires the data scientist to train the algorithm with both labeled inputs and desired outputs.

- State the set of assumptions (if any) related to the problem under consideration

  Presumptions are by using regression label encoding, data scaling, precision that it means the relationship between the dependent and independent variables look fairly linear. Thus, our linearity assumption is satisfied.

- Hardware and Software Requirements and Tools Used

By importing many libraries are

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pickle
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from matplotlib import pyplot
```

preprocessing, from sklearn.utlis import resample, label encoder enable_iterative_imputer, iterative_imputer, selectkbest, randomforestclassifier, mean_squared_error, mean_absolute_error, ridge, lasso, ridgecv, lassocv, qqplot, gradientboostingclassifier, standardscaler, confusion_matrix, classification_report, accuracy_score, gridsearchcv, kneighborsclassifier, logisticregression.

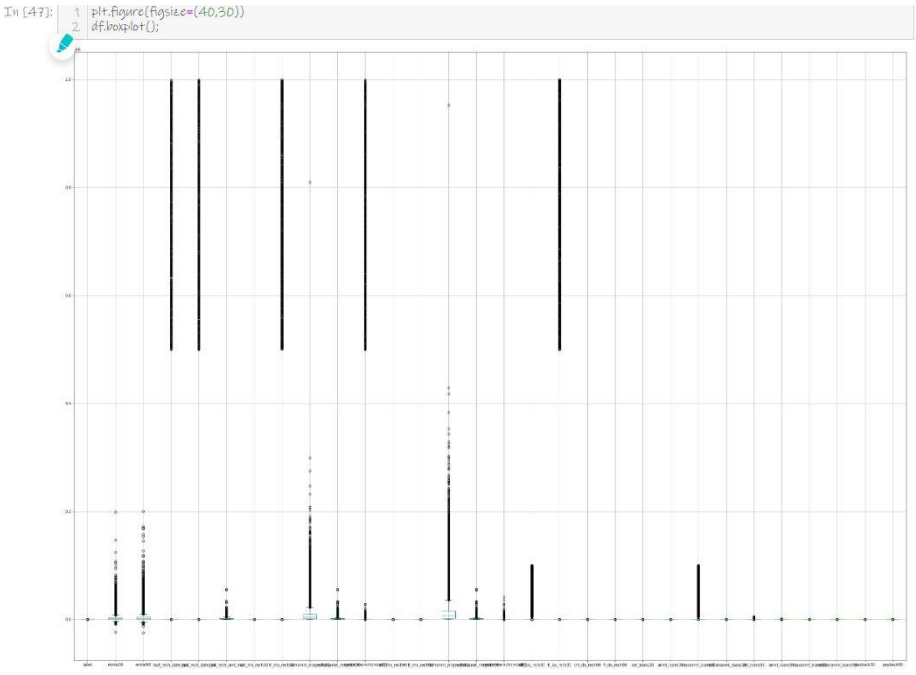
MODEL/s DEVELOPMENT AND EVALUATION

- Identification of possible problem-solving approaches (methods)

  The collection and interpretation of data in order to uncover patterns and trends. It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modelling or designing surveys and studies. The approaches/methods of identification are descriptive and inferential statistics which are describes as the properties of sample and population data, and inferential statistics which uses those properties to test hypotheses and draw efficient conclusions in terms of outputs.

- Testing of Identified Approaches (Algorithms)

OUTLIERS: [BOXPLOT]



REMOVE THE OUTLIERS:

OUTLIERS WITH IQR

In [120]:
```
Q1=MiceImputed.quantile(0.25)
Q3=MiceImputed.quantile(0.75)
IQR=Q3-Q1
print(IQR)
```

```
label                  0.00
rental30            3076.52
rental90            3901.53
last_rech_date_ma      6.00
last_rech_date_da      0.00
last_rech_amt_ma    1539.00
cnt_ma_rech30          4.00
fr_ma_rech30           6.00
sumamnt_ma_rech30    8470.00
medianamnt_ma_rech30 1154.00
medianmarechprebal30   72.00
cnt_ma_rech90          6.00
fr_ma_rech90           8.00
sumamnt_ma_rech90   13683.00
medianamnt_ma_rech90 1151.00
medianmarechprebal90   64.71
cnt_da_rech30          0.00
fr_da_rech30           0.00
cnt_da_rech90          0.00
fr_da_rech90           0.00
cnt_loans30            3.00
amnt_loans30          18.00
maxamnt_loans30        0.00
medianamnt_loans30     0.00
cnt_loans90            4.00
amnt_loans90          24.00
maxamnt_loans90        0.00
medianamnt_loans90     0.00
payback30              3.75
payback90              4.50
pcircle                0.00
pdate                 36.00
dtype: float64
```

In [121]:
```
type(df)
```

Out[121]: pandas.core.frame.DataFrame

REMOVE THE OUTLIERS

In [122]:
```
MiceImputed = MiceImputed[~((MiceImputed < (Q1 - 1.5 * IQR)) |(MiceImputed > (Q3 + 1.5 * IQR))).any(axis=1)]
MiceImputed.shape
```

Out[122]: (58174, 32)

- Run and Evaluate selected models

DATA SCALING:

```
In [129]:
from sklearn.preprocessing import StandardScaler
# Data Scaling Formula Z=(x-mean)/std
scaler = StandardScaler()
X_scaled=scaler.fit_transform(X)
X_scaled
```

```
Out[129]: array([[-0.23621479, -0.31934721, -0.13811893, ..., -0.85648223,
         0.        ,  2.24155209],
        [-0.11036683, -0.21530584,  0.149985  , ...,  0.02496435,
         0.        , -0.44262019],
        [ 0.64217573,  0.40683817, -0.7143268 , ...,  0.15088528,
         0.        , -0.76657202],
        ...,
        [-0.71720003, -0.71698862,  0.149985  , ..., -0.85648223,
         0.        , -1.36819684],
        [-0.11651456, -0.22038831, -0.7143268 , ..., -0.47871941,
         0.        , -0.67401435],
        [ 0.28816543,  0.11416982,  0.149985  , ..., -0.47871941,
         0.        , -0.90540852]])
```

```
In [130]:
regression=LinearRegression()
regression.fit(X_train,y_train)
```

```
Out[130]: LinearRegression
LinearRegression()
```

```
In [131]:
# Adjusted R2 score
regression.score(X_train,y_train)
```

```
Out[131]: 0.9555764509034571
```

```
In [132]:
regression.score(X_test,y_test)
```

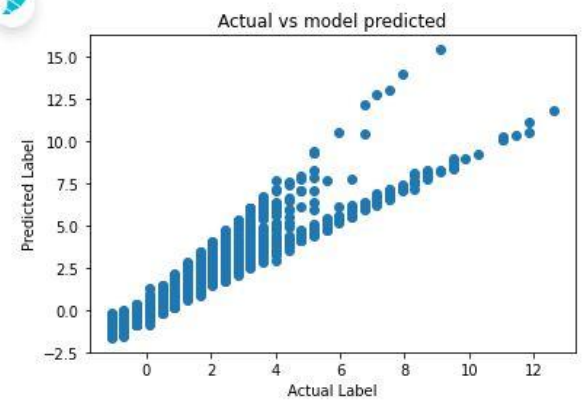
```
Out[132]: 0.9531666388767995
```

```
In [133]:
y_pred = regression.predict(X_test)
y_pred
```

```
Out[133]: array([ 0.45092892, -0.68693547, -0.6135218 , ...,  0.21967141,
        -0.26036992, -0.67005551])
```

```
In [134]:
plt.scatter(y_test,y_pred)
plt.xlabel('Actual Label')
plt.ylabel('Predicted Label')
plt.title('Actual vs model predicted')
plt.show
```

```
Out[134]: <function matplotlib.pyplot.show(close=None, block=None)>
```



MODEL EVALUATION:

MODEL EVALUATION

```
In [135]:
1 from sklearn.metrics import mean_squared_error, mean_absolute_error
2 y_pred=regression.predict(X_test)

In [136]:
1 mean_absolute_error(y_test,y_pred)
Out[136]: 0.10423435885709731

In [137]:
1 mean_squared_error(y_test,y_pred)
Out[137]: 0.04736428166250717

In [138]:
1 np.sqrt(mean_squared_error(y_test,y_pred))
Out[138]: 0.21763336523269397
```

GRADIENT BOOSTING CLASSIFIER: train and test result

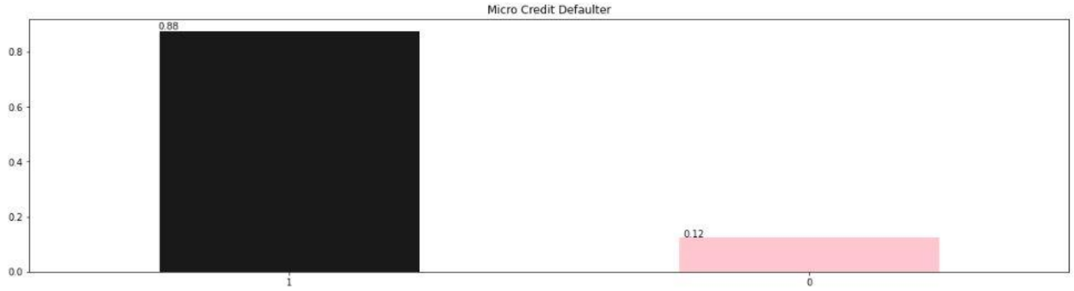
ACCURACY SCORE: 87%

```
====================Train Result=====================
Accuracy Score: 87.558191050676 48
_____________________________________________
CLASSIFICATION REPORT :
                    0           1  accuracy     macro avg  weighted avg
precision    1.000000    0.875576  0.875582    0.937788      0.891063
recall       0.000383    1.000000  0.875582    0.500192      0.875582
f1-score     0.000766    0.933661  0.875582    0.467214      0.817547
support  18261.000000  128454.000000  0.875582  146715.000000  146715.000000
_____________________________________________
Confusion Matrix:
 [[    7 18254]
 [    0 128454]]


==================Test Result=========================
Accuracy: 87.43280638697159
_____________________________________________
CLASSIFICATION REPORT :
                    0           1  accuracy     macro avg  weighted avg
precision     0.0     0.874342  0.874328     0.437171      0.764476
recall        0.0     0.999982  0.874328     0.499991      0.874328
f1-score      0.0     0.932951  0.874328     0.466475      0.815720
support    7901.0  54977.000000  0.874328  62878.000000  62878.000000
_____________________________________________
Confusion Matrix:
 [[    0  7901]
 [    1 54976]]
```

- Visualizations

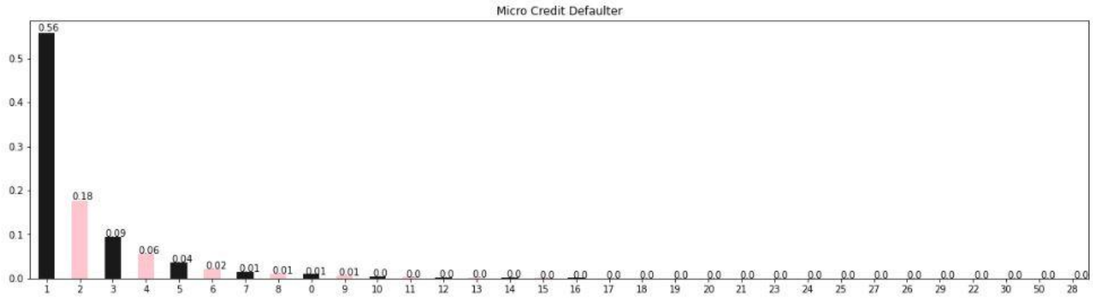
'LABEL'   MICRO CREDIT DEFAULTER

```python
In [111]:
import matplotlib.pyplot as plt
plt.figure(figsize=(20,5))
ax=df.label.value_counts(normalize=True).plot(kind='bar', color=['black', 'pink'], alpha=0.9, rot=0)
plt.title('Micro Credit Defaulter')
for i in ax.patches:
    ax.annotate(str(round(i.get_height(),2)),(i.get_x() * 1.01, i.get_height() * 1.01))

plt.show()
```



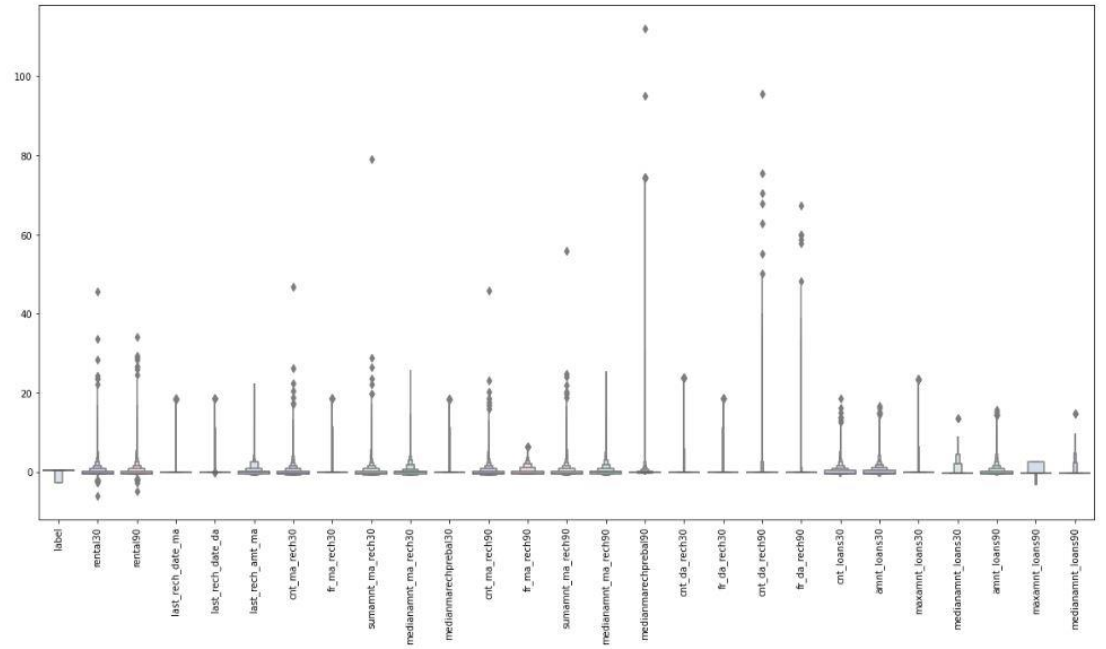
RESAMPLING THE LABEL

```python
In [112]:
from sklearn.utils import resample

failure=df[df.label == 0]
success=df[df.label == 1]
success_oversampled=resample(success, replace=True, n_samples=len(failure), random_state=42)
oversampled = pd.concat([failure, success_oversampled])

plt.figure(figsize=(20,5))

ax=oversampled.cnt_loans30.value_counts(normalize=True).plot(kind='bar', color=['black', 'pink'], alpha=0.9, rot=0)
plt.title('Micro Credit Defaulter')
for i in ax.patches:
    ax.annotate(str(round(i.get_height(),2)),(i.get_x() * 1.01, i.get_height() * 1.01))

plt.show()
```



BOXPLOT OF DATASETS

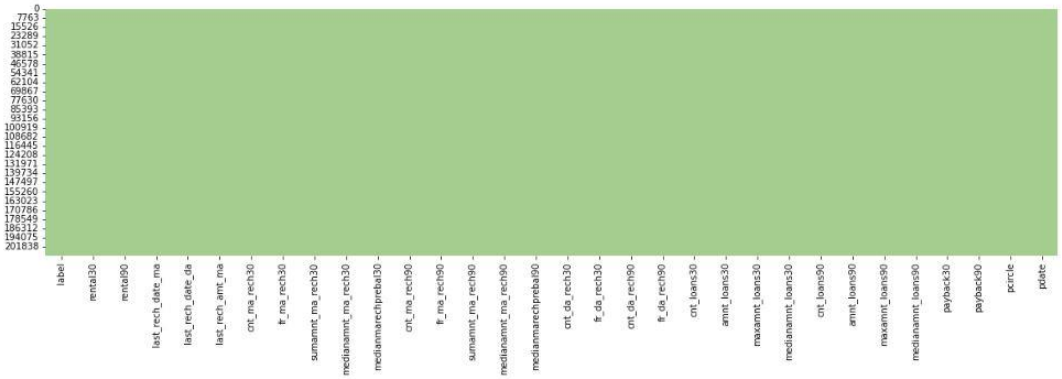
In [113]:
```
color=["#DDDBEE", "#C2C4E2", "#EED4E5", "#D1E6DC", "#BDE2E2"]
plt.figure(figsize=(20,10))
sns.boxenplot(data=features,palette=color)
plt.xticks(rotation=90)
plt.show()
```



USING THE HEATMAP FOR THE DATASETS

```
In [114]:
1 plt.figure(figsize=(20,5))
2 sns.heatmap(df.isnull(),cbar=False,cmap='crest')
```

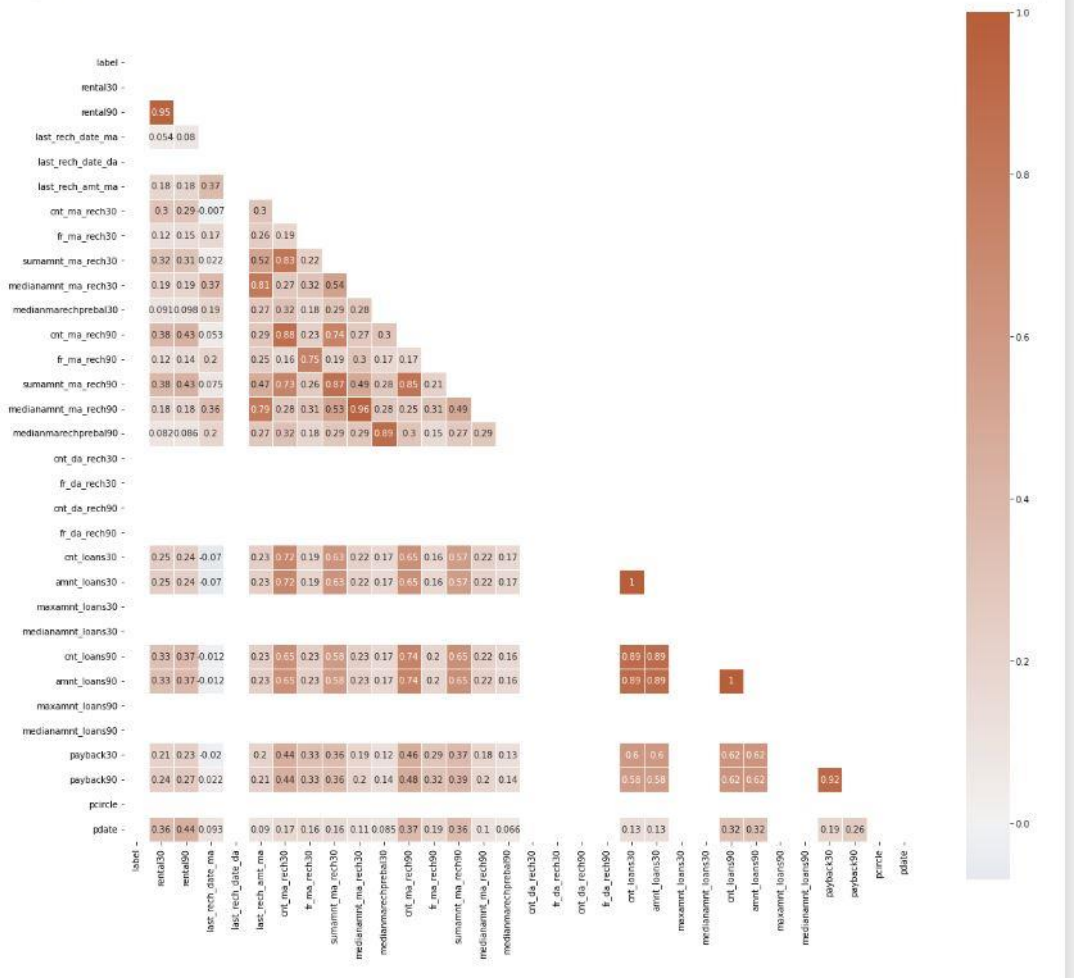
Out[114]: <AxesSubplot:>



AFTER REMOVING THE OUTLIERS

In [123]:

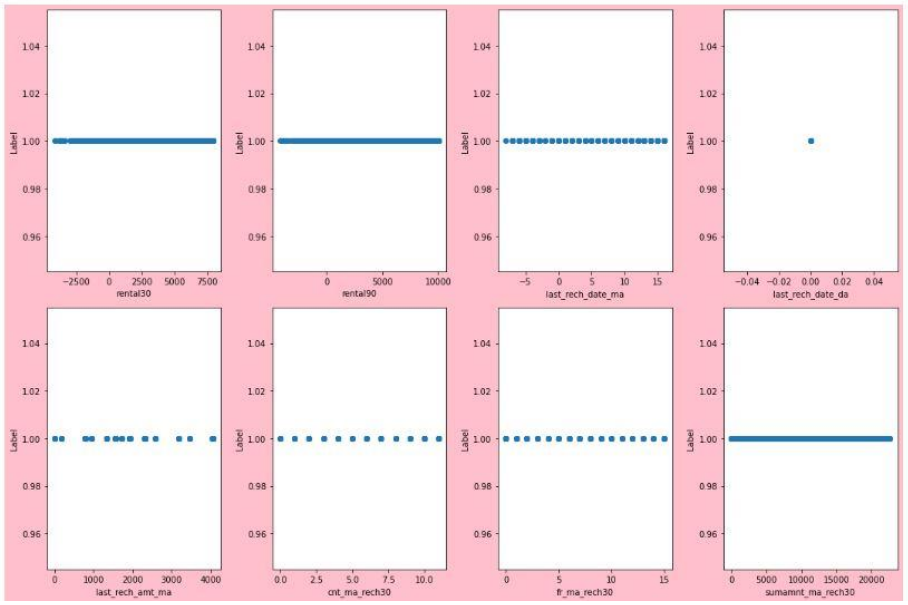
```
corr = WiceImputed.corr()
mask = np.triu(np.ones_like(corr, dtype=np.bool))
f, ax = plt.subplots(figsize=(20,20))
cmap = sns.diverging_palette(250, 25, as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=None, center=0,square=True, annot=True, linewidths=.5, cbar_kws={"shrink": .9})
plt.show()
```



VISUALIZING THE RELATIONSHIPS

In [128]:

```
# Visualizing relationship
plt.figure(figsize=(15,10), facecolor='pink')
plotnumber = 1

for column in X:
    if plotnumber<=8 :
        ax = plt.subplot(2,4,plotnumber)
        plt.scatter(X[column],y)
        plt.xlabel(column,fontsize=10)
        plt.ylabel('Label',fontsize=10)
    plotnumber+=1
plt.tight_layout()
```



The relationship between the dependent and independent variables look fairly linear. Thus, our linearity assumption is satisfied.

- Interpretation of the Results

  The results were interpreted from the visualizations, preprocessing and modelling:

1. The model of the independent variables and dependent variables are exactly vary with the variables.

2. It can accordingly manipulating the strategy of the areas that will yield high returns as it make easier for the institution in improving the client datasets in the micro-credit project.
3. It will take a good way for the management to understand the label of defaulter and non-defaulter of loan.
4. By doing visualization there are many things to be noted when it will according to work each other it means that from which aspect it is going to work when the defaulter and non-defaulter is the main target of the micro-credit case because according to this it will predicted.
5. By preprocessing the data it means that from the micro-credit case label encoder helps the dataset column to transform to fit another column in to it.

CONCLUSION

The institution provides small loans to poor clients who typically lack collateral, steady employment, verifiable credit history to improve borrowers from the selection of customers for the credit and the client wants some predictions that could help them in further investment and improvement in selection of customers.

I would like to conclude here that by doing the research on this project realized that the small creditors are at very large and its datasets are very lengthy the micro-finance institution has lot to do improve the condition and provide good facilities to the poor customers like Group Loans, Agricultural Loans, Individual Business Loans . By using data visualization elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. There are many things to be noted when it will according to work each other. By preprocessing the data it means that from the micro-credit case label encoder helps the dataset column to transform to fit another column in to it. The model of the project are ready to analyse the independent and dependent variable.

THANK YOU