Micro-Credit Defaulter Project

Submitted by:

Dolypona Das

ACKNOWLEDGMENT

I would like to thank our SMEs (Sajid Choudhary and Shubham Yadav) for their expert advice and encouragement throughout this difficult project, as well as some YouTube videos by KrishNaik and https://analyticsindiamag.com/.

INTRODUCTION

- Business Problem Framing

  In this project, we 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 5 days of insurance of loan.

  This project tells us about a Microfinance Institution (MFI) that offers financial services to low income populations. It becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income.

- Conceptual Background of the Domain Problem

  1. Firstly we will check whether the problem is supervised or not, mostly we will have a supervised one where there will be target variable.
  2. After that we will check whether the project is a regression type or a classification type.
  3. We will also check whether our dataset is balanced or imbalanced. If it is an imbalanced one, we will apply sampling techniques to balance the dataset.
  4. Then we will do model building and check its accuracy.
  5. Our main motto is to build a model with good accuracy and for that we will also go for hyperparameter tuning.

- Review of Literature

  I am summarizing my research done on the topic.

  - I have imported important libraries for my project.
  - I have created the dataframe.
  - I have analysed my data by checking its shape, number of columns, presence of null values if any and checking the datatypes.
  - Then I have done some data cleaning steps,e.g Checking the value counts of the target variable, dropping some irrelevant columns from the dataset, checking correlation between the

dependant and independent variables using heatmap, visualizing data using distribution plots, detecting and removing skewness in my data using power transform function, outliers detection using boxplots and removing them, balancing dataset using randomoversampler method, splitting the data into independent and dependant variables and finally scaling the data.

- Now I have used BalancedRandomForestClassifier for model buildng and I have applied the algorithm on the imbalanced dataset. I have seen 84% accuracy after hyperparameter tuning (RandomizedSearchCV).
- Then I have used DecisionTreeClassifier for model building. Now I have applied the algorithm on my balanced dataset achieved by RandomOverSampler and got accuracy of 94% after hyperparameter tuning(GridSearchCV) which is more satisfactory than the previous one.f1-score obtained is also quite good.

- Motivation for the Problem Undertaken

  The objective behind to make this project is to help especially the unbanked poor families living in remote areas with not much sources of income.

  I am motivated by microfinance institutions (MFI), experts and donorsare supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  If you look at data science, we are actually using mathematical models to model (and hopefully through the model to explain some of the things that we have seen) business circumstances, environment etc and through these model, we can get more insights

such as the outcomes of our decision undertaken, what should we do next or how shall we do it to improve the odds. So mathematical models are important, selecting the right one to answer the business question can bring tremendous value to the organization.
Here, I am using **Balanced Random Forest Classifier** on imbalanced datasets, that's the advantage of it and later can be stacked with other models. But I am not satisfied with the f1 score of the model, though accuracy is good.
Next, I have tried **Decision Tree Classifier** on the balanced datasets which I have achieved using RandomOverSampler and got good f1 score and the accuracy of the model is quite satisfactory.

- Data Sources and their formats

Data Source: The read_csv function of the pandas library is used to read the content of a CSV file into the python environment as a pandas DataFrame. The function can read the files from the OS by using proper path to the file.

Data description: Pandas describe() is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. When this method is applied to a series of string, it returns a different output which is shown below.

Out[22]:

	count	mean	std	min	25%	50%	75%	max
label	209593.0	0.875177	0.330519	0.000000	1.000	1.000000	1.00	1.00
daily_decr30	209593.0	5381.402289	9220.623400	-93.012667	42.440	1469.175667	7244.00	265926.00
daily_decr90	209593.0	6082.515068	10918.812767	-93.012667	42.692	1500.000000	7802.79	320630.00
rental30	209593.0	2692.581910	4308.586781	-23737.140000	280.420	1083.570000	3356.94	198926.11
rental90	209593.0	3483.406534	5770.461279	-24720.580000	300.260	1334.000000	4201.79	200148.11
last_rech_amt_ma	209593.0	2064.452797	2370.786034	0.000000	770.000	1539.000000	2309.00	55000.00
cnt_ma_rech30	209593.0	3.978057	4.256090	0.000000	1.000	3.000000	5.00	203.00
sumamnt_ma_rech30	209593.0	7704.501157	10139.621714	0.000000	1540.000	4628.000000	10010.00	810096.00
medianamnt_ma_rech30	209593.0	1812.817952	2070.864620	0.000000	770.000	1539.000000	1924.00	55000.00
cnt_ma_rech90	209593.0	6.315430	7.193470	0.000000	2.000	4.000000	8.00	336.00
fr_ma_rech90	209593.0	7.716780	12.590251	0.000000	0.000	2.000000	8.00	88.00
sumamnt_ma_rech90	209593.0	12396.218352	16857.793882	0.000000	2317.000	7226.000000	16000.00	953036.00
medianamnt_ma_rech90	209593.0	1864.595821	2081.680664	0.000000	773.000	1539.000000	1924.00	55000.00
medianmarechprebal90	209593.0	92.025541	369.215658	-200.000000	14.600	36.000000	79.31	41456.50
cnt_loans30	209593.0	2.758981	2.554502	0.000000	1.000	2.000000	4.00	50.00
amnt_loans30	209593.0	17.952021	17.379741	0.000000	6.000	12.000000	24.00	306.00
medianamnt_loans30	209593.0	0.054029	0.218039	0.000000	0.000	0.000000	0.00	3.00
amnt_loans90	209593.0	23.645398	26.469861	0.000000	6.000	12.000000	30.00	438.00
maxamnt_loans90	209593.0	6.703134	2.103864	0.000000	6.000	6.000000	6.00	12.00
medianamnt_loans90	209593.0	0.046077	0.200692	0.000000	0.000	0.000000	0.00	3.00
payback30	209593.0	3.398826	8.813729	0.000000	0.000	0.000000	3.75	171.50
payback90	209593.0	4.321485	10.308108	0.000000	0.000	1.666667	4.50	171.50

- Data Pre-processing Done
  - I have dropped the column ‘Unnamed:0’ as it is of no use, it is only giving serial numbers starting from 1.
  - Then I have checked the value counts of the target variable ’Label’ whether the dataset is balanced or imbalanced.
  - Again, I have dropped the column’ msisdn’, it tells us about the mobile number of the user which is not so important.
  - I have dropped ‘pcircle’ and ‘pdate’, pcircle is telling us about the telecom circle and pdate is giving us some dates, but these two columns are not playing any significant role.
  - I have checked the correlation between dependant and independent variables using heatmap. I have seen many columns which are 0% correlated with the target variable. So, I also have dropped all those columns which are not correlated with the target variable ‘Label’.
  - I have studied the statistical summary of the dataset.
  - I have done some visualization using distribution plots where I can clearly see some outliers and skewness from the plots.
  - I have checked outliers using boxplots and also removed them using zscore.
  - I have splitted the dependant and independent variables into x and y.
  - Then I have checked for the skewness and minimised the skewness using power transform function.
  - After that, I have visualized the target variable ‘Label’ using graph and can clearly see the imbalancing in the data.
  - I have installed imbalanced-learn and also updated scikit-learn.
  - Then I have performed RandomOversampler method to balance the dataset.
  - I have scaled the data using StandardScaler method and made my data ready for model building.

- Data Inputs- Logic- Output Relationships

- The logic between the data input and data output is very simple.
- Input:If the loan has been payed(non-defaulter),output is 1.
  If the loan has not been payed(defaulter),the output is 0.
- The model uses 21 features/attributes/columns from the data set to determine the output if it is 1 or 0.

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

I have made some assumptions like I have dropped some columns from the dataset as they are irrelevant from my point of view.

After checking the correlation using heatmap, I have seen there are some columns which are 0% correlated with the target variable, so I have dropped those columns to make the data clean.

- Hardware and Software Requirements and Tools Used
  - Hardware requirements:
    Processor: Intel(R) Celeron(R) CPU N3050 @1.60 GHz 1.60 GHz
    RAM: 3.92 GB
    System type: 64-bit operating system, x64-based processor
  - Software requirements:
    Python: One of the most used programming languages
  - Tools used:
    Jupyter notebook: Jupyter is a free, open-source, interactive web tool known as a computational notebook where I have written my python codes.
    NumPy: NumPy is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures.
    Pandas: Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.

Matplotlib: It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python.

Seaborn: It is also a Python library used for plotting graphs with the help of Matplotlib, Pandas, and Numpy.

Scikit-learn: It is probably the most useful library for machine learning in Python.The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

Scipy.stats: This module contains a large number of probability distributions as well as a growing library of statistical functions.

Imbalanced-learn: Imbalanced-learn (imported as **imblearn**) is an open source, MIT-licensed library relying on scikit-learn (imported as sklearn) and provides tools when dealing with classification with imbalanced classes.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  1. To check the correlation among the data,I have used **heatmap** to visualize it.

  2. To get a clear view of the columns visually,I have used **distribution plots**.

  3. For checking outliers,I have used **boxplots** and, **zscore method** to remove those outliers.

  4. For skewness removal ,I have used **power transform function** in my project.

5. For balancing the dataset, I have used **Random Over Sampler method.**

6. For scaling the data, I have used **StandardScaler** method.

7. For training and testing the data, I have imported **train_test_split library** from scikit-learn.

8.For model building, I have used **BalancedRandomForestClassifier** on my imblanced data and **Decision Tree Classifier** on my balanced data.

8. For better accuracy of the model, I have used **hyperparameter tuning (RandomizedSearchCV and GridSearchCV).**

- Testing of Identified Approaches (Algorithms)

I have used two algorithms for testing .

1. Balanced Random Forest Classifier: I have chosen the random forest classifier because this algorithm is applied easily on the imbalanced dataset. One of the biggest advantages of random forest is its versatility. It can be used for both regression and classification tasks, and it's also easy to view the relative importance it assigns to the input features.

2. Decision Tree Classifier: When I have seen the result of the previous algorithm, the accuracy and the f1 score can be improved, so I have chosen the decision tree classifier but the algorithm was applied on the balanced dataset. One of the advantages of decision trees is that their outputs are easy to read and interpret, without even requiring statistical knowledge, they are easy to prepare and less data cleaning is required.

- Run and Evaluate selected models

<u>Balanced Random Forest Classifier:</u>

```
In [47]:
#Model building
#I will be using Balanced RandomForestClassifier which  can be used directly on imbalanced datasets, that's the advantage an
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
from imblearn.ensemble import BalancedRandomForestClassifier
brf = BalancedRandomForestClassifier()
brf.fit(x_train,y_train)
brf.score(x_train,y_train)

Out[47]: 0.8900499430180331

In [48]:
brf.score(x_test,y_test)

Out[48]: 0.8312836054355265
```

The score for the training dataset is 89% and the testing dataset is 83%.

The metrics that I have used is given in the screenshot below.

```
In [56]:
brf=BalancedRandomForestClassifier(n_estimators=900,min_samples_split=10,min_samples_leaf=1,max_features='auto',max_depth=30
brf.fit(x_train,y_train)
brf.score(x_train,y_train)
pred=brf.predict(x_test)
print(accuracy_score(y_test,pred))
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
0.8371492814546877
[[ 5571  1590]
 [ 6739 37245]]
              precision    recall  f1-score   support

           0       0.45      0.78      0.57      7161
           1       0.96      0.85      0.90     43984

    accuracy                           0.84     51145
   macro avg       0.71      0.81      0.74     51145
weighted avg       0.89      0.84      0.85     51145
```

From the snapshot, we can see the accuracy and f1 score which should be improved.

<u>Decision Tree Classifier</u>:

```
In [57]:
x_res_train,x_res_test,y_res_train,y_res_test = train_test_split(x_res,y_res,test_size=0.3)
```

```
In [58]:
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(x_res_train,y_res_train)
dtc.score(x_res_train,y_res_train)
```

```
Out[58]: 0.9974892099283257
```

```
In [59]:
dtc.score(x_res_test,y_res_test)
```

```
Out[59]: 0.9440035423549848
```

The score for the training balanced data is 100% and testing data is 94%.

```
In [65]:
dtc=DecisionTreeClassifier(criterion='gini')
dtc.fit(x_res_train,y_res_train)
dtc.score(x_res_train,y_res_train)
pred=dtc.predict(x_res_test)
print(accuracy_score(y_res_test,pred))
print(confusion_matrix(y_res_test,pred))
print(classification_report(y_res_test,pred))
```

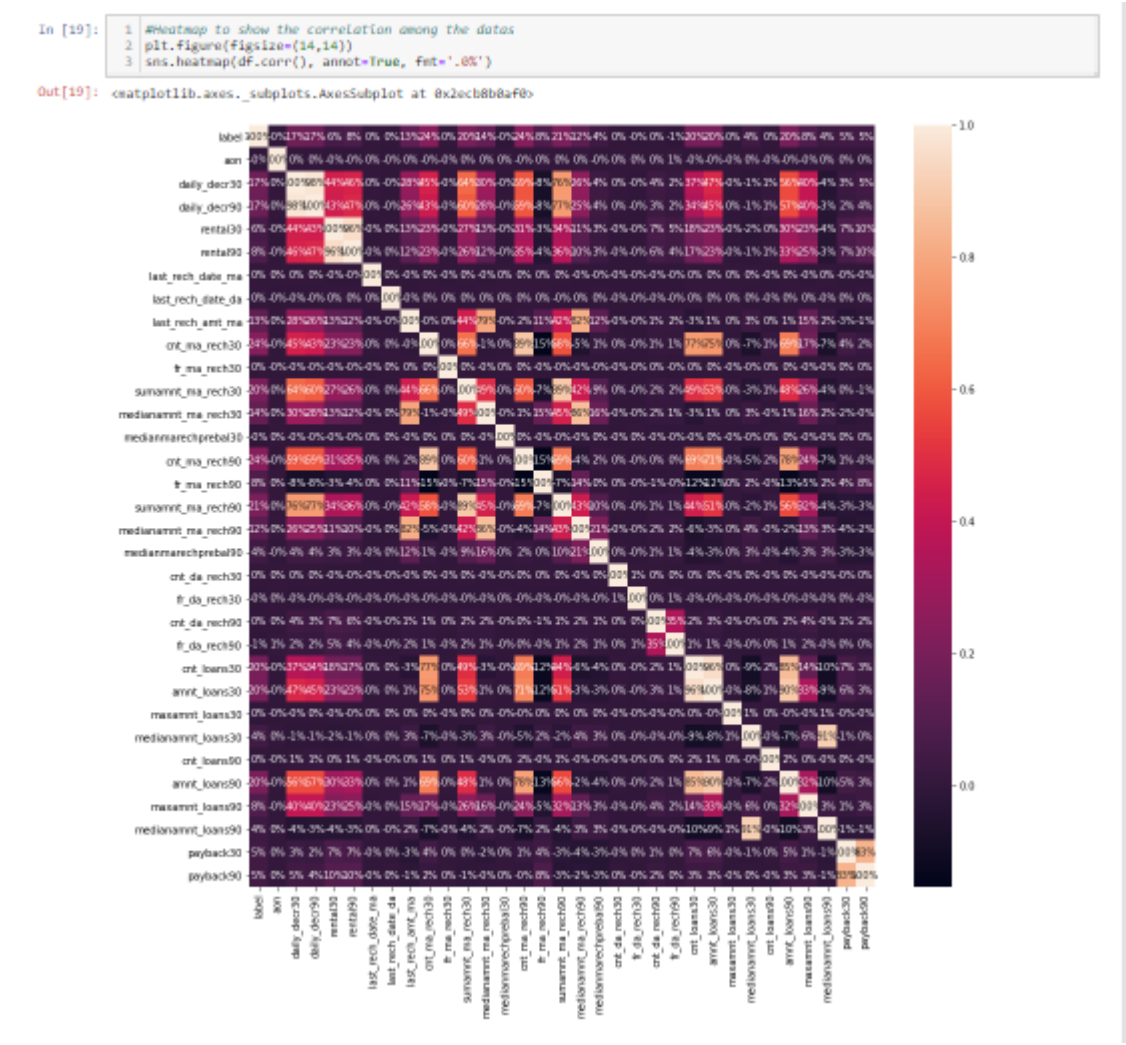
```
0.9441624941812278
[[43743   258]
 [ 4660 39416]]
              precision    recall  f1-score   support

           0       0.90      0.99      0.95     44001
           1       0.99      0.89      0.94     44076

    accuracy                           0.94     88077
   macro avg       0.95      0.94      0.94     88077
weighted avg       0.95      0.94      0.94     88077
```

The accuracy obtained after hyperparametertuning is 94% and the f1 score is 95% for ‘label’ 0 and 94% for ‘label’ 1 which is improved than the previous one.

- Key Metrics for success in solving problem under consideration

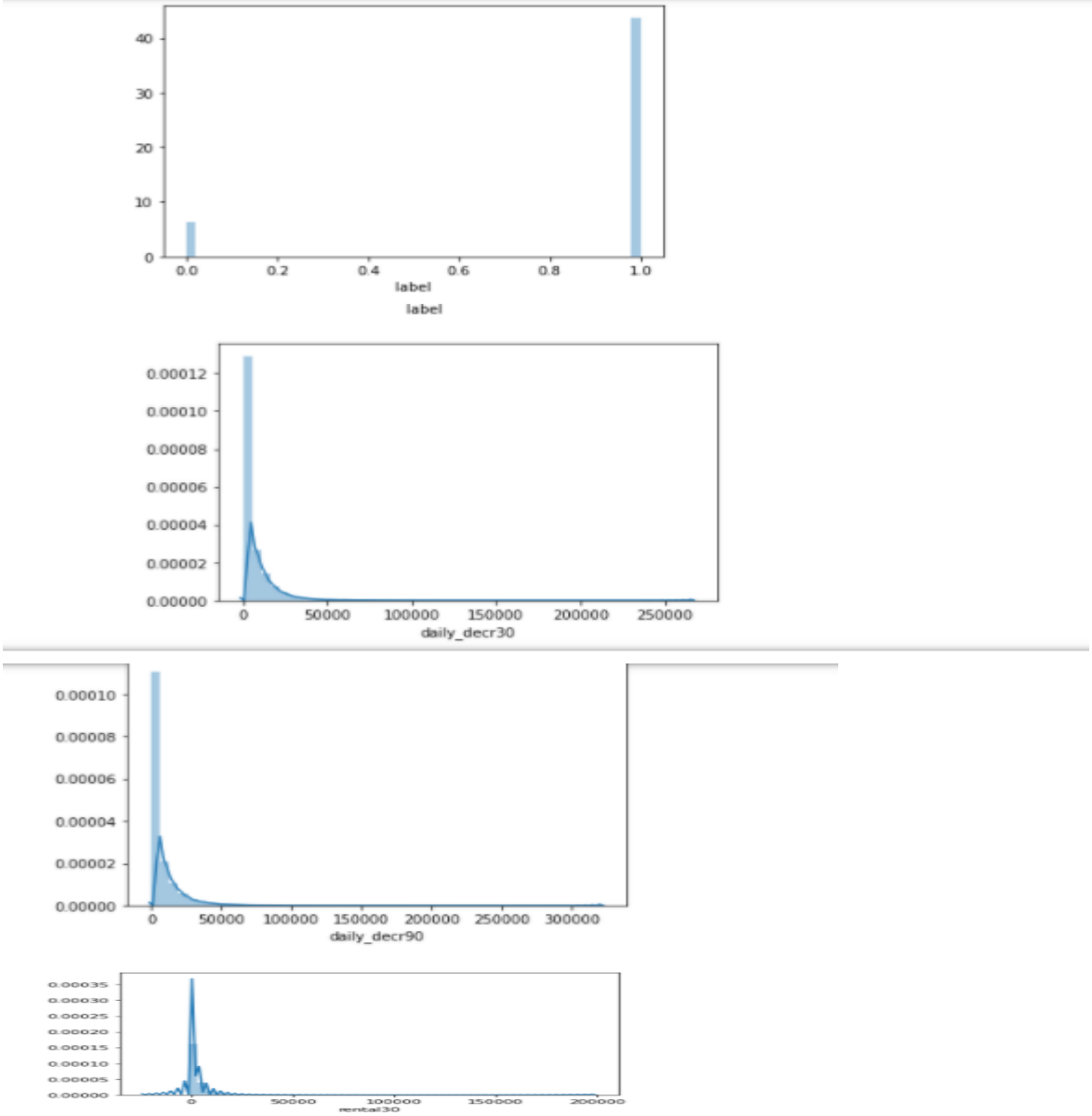
The key metrics used are accuracy_score,confusion_matrix and classification_report for both the algorithms.

- Visualizations



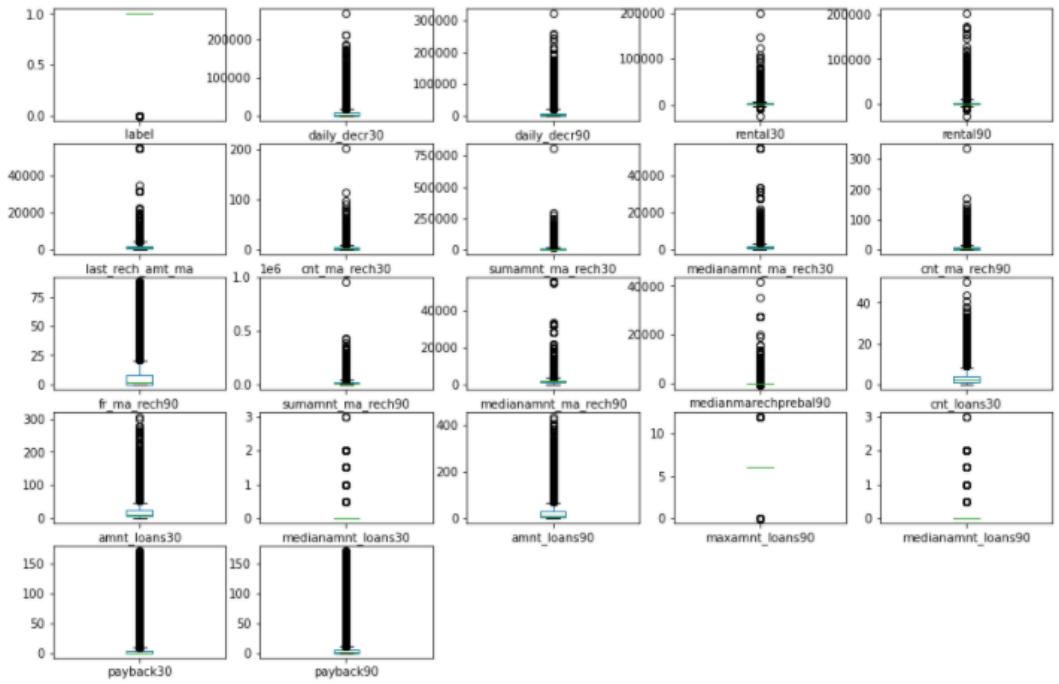
This is the heatmap where I have checked the correlation between the data and also got to know that there are columns or independent variables which are 0% correlated with the target variable.

```
In [25]:
columns=['label', 'daily_decr30', 'daily_decr90', 'rental30', 'rental90',
       'last_rech_amt_ma', 'cnt_ma_rech30', 'sumamnt_ma_rech30',
       'medianamnt_ma_rech30', 'cnt_ma_rech90', 'fr_ma_rech90',
       'sumamnt_ma_rech90', 'medianamnt_ma_rech90', 'medianmarechprebal90',
       'cnt_loans30', 'amnt_loans30', 'medianamnt_loans30', 'amnt_loans90',
       'maxamnt_loans90', 'medianamnt_loans90', 'payback30', 'payback90']
for i in df[columns]:
    plt.figure()
    sns.distplot(df[i])
```



These are some distribution plots of the columns of our dataset where I can see how my data looks like.

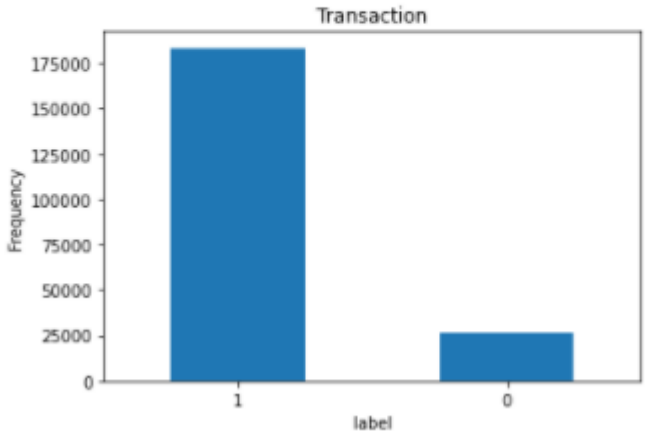
```
In [26]:
#Checking outliers using boxplots
df.plot(kind='box',subplots=True,layout=(5,5),figsize=(15,10));
```



The above screenshot is of the boxplots where I can clearly see there are lots of outliers present in our data.

```
In [37]:
count_classes=pd.value_counts(df['label'],sort=True)
count_classes.plot(kind='bar',rot=0)
plt.title('Transaction')
plt.xlabel('label')
plt.ylabel('Frequency')
```

```
Out[37]: Text(0, 0.5, 'Frequency')
```



The barplot is giving a clear view of the target variable ‘label’ which is imbalanced.

- Interpretation of the Results

In the visualization part, I have seen how my data looks like using heatmap, boxplot, distribution plots, bar plot etc.

In the pre-processing part, I have cleaned my data using many methods like zscore, power_transform function etc.

In the modelling part, I have designed our model using algorithms like Balanced Random Forest Classifier and Decision Tree Classifier. The accuracy ,f1 score, confusion_matrix, classification_report are achieved for each model.

CONCLUSION

- Key Findings and Conclusions of the Study

  The key findings are we have to study the data very clearly so that we are able to decide which data are relevant for our findings. The techniques that I have used are heatmap, zscore, power_transform function etc.

  The conclusion of our study is we have to achieve a model with good accuracy and f1-score.

- Learning Outcomes of the Study in respect of Data Science

  We will develop relevant programming abilities. We will demonstrate proficiency with statistical analysis of data. We will develop the ability to build and assess data-based models. We will execute statistical analyses with professional statistical software.

  The best algorithm for this project according to my work is Decision Tree Classifier because the accuracy and f1 score that I have achieved is quite satisfactory than the other model.

- Limitations of this work and Scope for Future Work

  This study has proposed a comprehensive research and model development for the prediction of the default loans. As the issue related to the high ratio of bad loans is very much critical especially in micro-financing banks of various under develop and developed countries. Although, loan lending has been proven very substantial

in the stability of any country's economy in this century such a huge amount of loan defaults is also very critical. To cope up with this problem a comprehensive amount of literature was reviewed to study the significant factors that lead to such problems.

Future scope of this work is we can try different algorithms like Logistic Regression, Gaussian NB etc for model building and try to achieve a good accuracy and f1-score.