

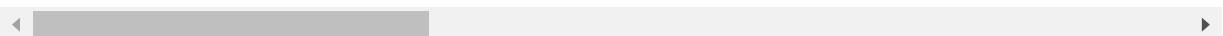
```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: df = pd.read_csv (r'C:/Users/Althaf/Downloads/Micro-Credit-Project/Data file.csv')
df
```

Out[2]:

| | | Unnamed: 0 | label | msisdn | aon | daily_decr30 | daily_decr90 | rental30 | rental90 | I: |
|--|---------------|---------------|-------|---------------|--------|--------------|--------------|----------|----------|-----|
| | | 0 | 1 | 0 21408170789 | 272.0 | 3055.050000 | 3065.150000 | 220.13 | 260.13 | |
| | | 1 | 2 | 1 76462170374 | 712.0 | 12122.000000 | 12124.750000 | 3691.26 | 3691.26 | |
| | | 2 | 3 | 1 17943170372 | 535.0 | 1398.000000 | 1398.000000 | 900.13 | 900.13 | |
| | | 3 | 4 | 1 55773170781 | 241.0 | 21.228000 | 21.228000 | 159.42 | 159.42 | |
| | | 4 | 5 | 1 03813182730 | 947.0 | 150.619333 | 150.619333 | 1098.90 | 1098.90 | |
| | | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| | 209588 | 209589 | 1 | 22758185348 | 404.0 | 151.872333 | 151.872333 | 1089.19 | 1089.19 | |
| | 209589 | 209590 | 1 | 95583184455 | 1075.0 | 36.936000 | 36.936000 | 1728.36 | 1728.36 | |
| | 209590 | 209591 | 1 | 28556185350 | 1013.0 | 11843.111670 | 11904.350000 | 5861.83 | 8893.20 | |
| | 209591 | 209592 | 1 | 59712182733 | 1732.0 | 12488.228330 | 12574.370000 | 411.83 | 984.58 | |
| | 209592 | 209593 | 1 | 65061185339 | 1581.0 | 4489.362000 | 4534.820000 | 483.92 | 631.20 | |

209593 rows × 37 columns



```
In [3]: df=pd.read_csv(r'C:/Users/Althaf/Downloads/Micro-Credit-Project/Data file.csv'  
,index_col=['Unnamed: 0'])  
print(df)
```

| | label | msisdn | aon | daily_decr30 | daily_decr90 | rental30 | \ |
|--------|-----------------|--------------------|-------------------|------------------|--------------|----------|-----|
| 1 | 0 | 21408I70789 | 272.0 | 3055.050000 | 3065.150000 | 220.13 | |
| 2 | 1 | 76462I70374 | 712.0 | 12122.000000 | 12124.750000 | 3691.26 | |
| 3 | 1 | 17943I70372 | 535.0 | 1398.000000 | 1398.000000 | 900.13 | |
| 4 | 1 | 55773I70781 | 241.0 | 21.228000 | 21.228000 | 159.42 | |
| 5 | 1 | 03813I82730 | 947.0 | 150.619333 | 150.619333 | 1098.90 | |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 209589 | 1 | 22758I85348 | 404.0 | 151.872333 | 151.872333 | 1089.19 | |
| 209590 | 1 | 95583I84455 | 1075.0 | 36.936000 | 36.936000 | 1728.36 | |
| 209591 | 1 | 28556I85350 | 1013.0 | 11843.111670 | 11904.350000 | 5861.83 | |
| 209592 | 1 | 59712I82733 | 1732.0 | 12488.228330 | 12574.370000 | 411.83 | |
| 209593 | 1 | 65061I85339 | 1581.0 | 4489.362000 | 4534.820000 | 483.92 | |
| | | | | | | | |
| | rental90 | last_rech_date_ma | last_rech_date_da | last_rech_amt_ma | ... | | |
| \ | | | | | | | |
| 1 | 260.13 | 2.0 | 0.0 | 1539 | ... | | |
| 2 | 3691.26 | 20.0 | 0.0 | 5787 | ... | | |
| 3 | 900.13 | 3.0 | 0.0 | 1539 | ... | | |
| 4 | 159.42 | 41.0 | 0.0 | 947 | ... | | |
| 5 | 1098.90 | 4.0 | 0.0 | 2309 | ... | | |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 209589 | 1089.19 | 1.0 | 0.0 | 4048 | ... | | |
| 209590 | 1728.36 | 4.0 | 0.0 | 773 | ... | | |
| 209591 | 8893.20 | 3.0 | 0.0 | 1539 | ... | | |
| 209592 | 984.58 | 2.0 | 38.0 | 773 | ... | | |
| 209593 | 631.20 | 13.0 | 0.0 | 7526 | ... | | |
| | | | | | | | |
| | maxamnt_loans30 | medianamnt_loans30 | cnt_loans90 | amnt_loans90 | \ | | |
| 1 | 6.0 | 0.0 | 2.0 | 12 | | | |
| 2 | 12.0 | 0.0 | 1.0 | 12 | | | |
| 3 | 6.0 | 0.0 | 1.0 | 6 | | | |
| 4 | 6.0 | 0.0 | 2.0 | 12 | | | |
| 5 | 6.0 | 0.0 | 7.0 | 42 | | | |
| ... | ... | ... | ... | ... | ... | | |
| 209589 | 6.0 | 0.0 | 2.0 | 12 | | | |
| 209590 | 6.0 | 0.0 | 3.0 | 18 | | | |
| 209591 | 12.0 | 0.0 | 6.0 | 54 | | | |
| 209592 | 12.0 | 0.0 | 3.0 | 24 | | | |
| 209593 | 12.0 | 0.0 | 2.0 | 18 | | | |
| | | | | | | | |
| | maxamnt_loans90 | medianamnt_loans90 | payback30 | payback90 | pcircle | \ | |
| 1 | 6 | 0.0 | 29.000000 | 29.000000 | UPW | | |
| 2 | 12 | 0.0 | 0.000000 | 0.000000 | UPW | | |
| 3 | 6 | 0.0 | 0.000000 | 0.000000 | UPW | | |
| 4 | 6 | 0.0 | 0.000000 | 0.000000 | UPW | | |
| 5 | 6 | 0.0 | 2.333333 | 2.333333 | UPW | | |
| ... | ... | ... | ... | ... | ... | ... | |
| 209589 | 6 | 0.0 | 1.000000 | 1.000000 | UPW | | |
| 209590 | 6 | 0.0 | 1.000000 | 1.000000 | UPW | | |
| 209591 | 12 | 0.0 | 4.000000 | 3.833333 | UPW | | |
| 209592 | 12 | 0.0 | 0.000000 | 10.500000 | UPW | | |
| 209593 | 12 | 0.0 | 0.000000 | 0.000000 | UPW | | |
| | | | | | | | |
| | pdate | | | | | | |
| 1 | 20-07-2016 | | | | | | |
| 2 | 10-08-2016 | | | | | | |
| 3 | 19-08-2016 | | | | | | |

```
4      06-06-2016
5      22-06-2016
...
209589 17-06-2016
209590 12-06-2016
209591 29-07-2016
209592 25-07-2016
209593 07-07-2016
```

```
[209593 rows x 36 columns]
```

```
In [4]: df.shape
```

```
Out[4]: (209593, 36)
```

```
In [5]: df.columns
```

```
Out[5]: Index(['label', 'msisdn', 'aon', 'daily_decr30', 'daily_decr90', 'rental30',
       'rental90', 'last_rech_date_ma', 'last_rech_date_da',
       'last_rech_amt_ma', 'cnt_ma_rech30', 'fr_ma_rech30',
       'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'medianmarechprebal30',
       'cnt_ma_rech90', 'fr_ma_rech90', 'sumamnt_ma_rech90',
       'medianamnt_ma_rech90', 'medianmarechprebal90', 'cnt_da_rech30',
       'fr_da_rech30', 'cnt_da_rech90', 'fr_da_rech90', 'cnt_loans30',
       'amnt_loans30', 'maxamnt_loans30', 'medianamnt_loans30', 'cnt_loans9
       0',
       'amnt_loans90', 'maxamnt_loans90', 'medianamnt_loans90', 'payback30',
       'payback90', 'pcircle', 'pdate'],
      dtype='object')
```

```
In [6]: df.dtypes
```

```
Out[6]: label          int64
msisdn         object
aon           float64
daily_decr30   float64
daily_decr90   float64
rental30       float64
rental90       float64
last_rech_date_ma float64
last_rech_date_da float64
last_rech_amt_ma  int64
cnt_ma_rech30    int64
fr_ma_rech30     float64
sumamnt_ma_rech30 float64
medianamnt_ma_rech30 float64
medianmarechprebal30 float64
cnt_ma_rech90    int64
fr_ma_rech90     int64
sumamnt_ma_rech90 int64
medianamnt_ma_rech90 float64
medianmarechprebal90 float64
cnt_da_rech30    float64
fr_da_rech30     float64
cnt_da_rech90    int64
fr_da_rech90     int64
cnt_loans30      int64
amnt_loans30     int64
maxamnt_loans30  float64
medianamnt_loans30 float64
cnt_loans90      float64
amnt_loans90     int64
maxamnt_loans90  int64
medianamnt_loans90 float64
payback30        float64
payback90        float64
pcircle          object
pdate            object
dtype: object
```

In [7]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 209593 entries, 1 to 209593
Data columns (total 36 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   label            209593 non-null   int64  
 1   msisdn          209593 non-null   object  
 2   aon              209593 non-null   float64 
 3   daily_decr30    209593 non-null   float64 
 4   daily_decr90    209593 non-null   float64 
 5   rental30         209593 non-null   float64 
 6   rental90         209593 non-null   float64 
 7   last_rech_date_ma 209593 non-null   float64 
 8   last_rech_date_da 209593 non-null   float64 
 9   last_rech_amt_ma 209593 non-null   int64  
 10  cnt_ma_rech30   209593 non-null   int64  
 11  fr_ma_rech30   209593 non-null   float64 
 12  sumamnt_ma_rech30 209593 non-null   float64 
 13  medianamnt_ma_rech30 209593 non-null   float64 
 14  medianmarechprebal30 209593 non-null   float64 
 15  cnt_ma_rech90   209593 non-null   int64  
 16  fr_ma_rech90   209593 non-null   int64  
 17  sumamnt_ma_rech90 209593 non-null   int64  
 18  medianamnt_ma_rech90 209593 non-null   float64 
 19  medianmarechprebal90 209593 non-null   float64 
 20  cnt_da_rech30   209593 non-null   float64 
 21  fr_da_rech30   209593 non-null   float64 
 22  cnt_da_rech90   209593 non-null   int64  
 23  fr_da_rech90   209593 non-null   int64  
 24  cnt_loans30    209593 non-null   int64  
 25  amnt_loans30   209593 non-null   int64  
 26  maxamnt_loans30 209593 non-null   float64 
 27  medianamnt_loans30 209593 non-null   float64 
 28  cnt_loans90    209593 non-null   float64 
 29  amnt_loans90   209593 non-null   int64  
 30  maxamnt_loans90 209593 non-null   int64  
 31  medianamnt_loans90 209593 non-null   float64 
 32  payback30      209593 non-null   float64 
 33  payback90      209593 non-null   float64 
 34  pcircle         209593 non-null   object  
 35  pdate           209593 non-null   object  
dtypes: float64(21), int64(12), object(3)
memory usage: 59.2+ MB
```

In [8]: `df.describe()`

Out[8]:

| | label | aon | daily_decr30 | daily_decr90 | rental30 | rental90 |
|--------------|---------------|---------------|---------------|---------------|---------------|---------------|
| count | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000 | 209593.000000 |
| mean | 0.875177 | 8112.343445 | 5381.402289 | 6082.515068 | 2692.581910 | 3483.40651 |
| std | 0.330519 | 75696.082531 | 9220.623400 | 10918.812767 | 4308.586781 | 5770.46121 |
| min | 0.000000 | -48.000000 | -93.012667 | -93.012667 | -23737.140000 | -24720.58000 |
| 25% | 1.000000 | 246.000000 | 42.440000 | 42.692000 | 280.420000 | 300.26000 |
| 50% | 1.000000 | 527.000000 | 1469.175667 | 1500.000000 | 1083.570000 | 1334.00000 |
| 75% | 1.000000 | 982.000000 | 7244.000000 | 7802.790000 | 3356.940000 | 4201.79000 |
| max | 1.000000 | 999860.755200 | 265926.000000 | 320630.000000 | 198926.110000 | 200148.11000 |

8 rows × 33 columns

In [9]: `df.drop(['msisdn', 'pcircle', 'pdate'], axis=1, inplace=True)`
`df`

Out[9]:

| | label | aon | daily_decr30 | daily_decr90 | rental30 | rental90 | last_rech_date_ma | last_re |
|---------------|-------|--------|--------------|--------------|----------|----------|-------------------|---------|
| 1 | 0 | 272.0 | 3055.050000 | 3065.150000 | 220.13 | 260.13 | | 2.0 |
| 2 | 1 | 712.0 | 12122.000000 | 12124.750000 | 3691.26 | 3691.26 | | 20.0 |
| 3 | 1 | 535.0 | 1398.000000 | 1398.000000 | 900.13 | 900.13 | | 3.0 |
| 4 | 1 | 241.0 | 21.228000 | 21.228000 | 159.42 | 159.42 | | 41.0 |
| 5 | 1 | 947.0 | 150.619333 | 150.619333 | 1098.90 | 1098.90 | | 4.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 209589 | 1 | 404.0 | 151.872333 | 151.872333 | 1089.19 | 1089.19 | | 1.0 |
| 209590 | 1 | 1075.0 | 36.936000 | 36.936000 | 1728.36 | 1728.36 | | 4.0 |
| 209591 | 1 | 1013.0 | 11843.111670 | 11904.350000 | 5861.83 | 8893.20 | | 3.0 |
| 209592 | 1 | 1732.0 | 12488.228330 | 12574.370000 | 411.83 | 984.58 | | 2.0 |
| 209593 | 1 | 1581.0 | 4489.362000 | 4534.820000 | 483.92 | 631.20 | | 13.0 |

209593 rows × 33 columns

In [10]: `from scipy.stats import zscore`
`z_score=abs(zscore(df))`
`print(df.shape)`
`df_final=df.loc[(z_score<3).all(axis=1)]`
`print(df_final.shape)`

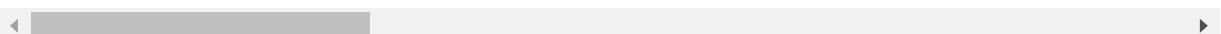
(209593, 33)
(161465, 33)

```
In [11]: dfcor=df.corr()  
dfcor
```

Out[11]:

| | label | aon | daily_decr30 | daily_decr90 | rental30 | rental90 | k |
|-----------------------------|--------------|------------|---------------------|---------------------|-----------------|-----------------|----------|
| label | 1.000000 | -0.003785 | 0.168298 | 0.166150 | 0.058085 | 0.075521 | |
| aon | -0.003785 | 1.000000 | 0.001104 | 0.000374 | -0.000960 | -0.000790 | |
| daily_decr30 | 0.168298 | 0.001104 | 1.000000 | 0.977704 | 0.442066 | 0.458977 | |
| daily_decr90 | 0.166150 | 0.000374 | 0.977704 | 1.000000 | 0.434685 | 0.471730 | |
| rental30 | 0.058085 | -0.000960 | 0.442066 | 0.434685 | 1.000000 | 0.955237 | |
| rental90 | 0.075521 | -0.000790 | 0.458977 | 0.471730 | 0.955237 | 1.000000 | |
| last_rech_date_ma | 0.003728 | 0.001692 | 0.000487 | 0.000908 | -0.001095 | -0.001688 | |
| last_rech_date_da | 0.001711 | -0.001693 | -0.001636 | -0.001886 | 0.003261 | 0.002794 | |
| last_rech_amt_ma | 0.131804 | 0.004256 | 0.275837 | 0.264131 | 0.127271 | 0.121416 | |
| cnt_ma_rech30 | 0.237331 | -0.003148 | 0.451385 | 0.426707 | 0.233343 | 0.230260 | |
| fr_ma_rech30 | 0.001330 | -0.001163 | -0.000577 | -0.000343 | -0.001219 | -0.000503 | |
| sumamnt_ma_rech30 | 0.202828 | 0.000707 | 0.636536 | 0.603886 | 0.272649 | 0.259709 | |
| medianamnt_ma_rech30 | 0.141490 | 0.004306 | 0.295356 | 0.282960 | 0.129853 | 0.120242 | |
| medianmarechprebal30 | -0.004829 | 0.003930 | -0.001153 | -0.000746 | -0.001415 | -0.001237 | |
| cnt_ma_rech90 | 0.236392 | -0.002725 | 0.587338 | 0.593069 | 0.312118 | 0.345293 | |
| fr_ma_rech90 | 0.084385 | 0.004401 | -0.078299 | -0.079530 | -0.033530 | -0.036524 | |
| sumamnt_ma_rech90 | 0.205793 | 0.001011 | 0.762981 | 0.768817 | 0.342306 | 0.360601 | |
| medianamnt_ma_rech90 | 0.120855 | 0.004909 | 0.257847 | 0.250518 | 0.110356 | 0.103151 | |
| medianmarechprebal90 | 0.039300 | -0.000859 | 0.037495 | 0.036382 | 0.027170 | 0.029547 | |
| cnt_da_rech30 | 0.003827 | 0.001564 | 0.000700 | 0.000661 | -0.001105 | -0.000548 | |
| fr_da_rech30 | -0.000027 | 0.000892 | -0.001499 | -0.001570 | -0.002558 | -0.002345 | |
| cnt_da_rech90 | 0.002999 | 0.001121 | 0.038814 | 0.031155 | 0.072255 | 0.056282 | |
| fr_da_rech90 | -0.005418 | 0.005395 | 0.020673 | 0.016437 | 0.046761 | 0.036886 | |
| cnt_loans30 | 0.196283 | -0.001826 | 0.366116 | 0.340387 | 0.180203 | 0.171595 | |
| amnt_loans30 | 0.197272 | -0.001726 | 0.471492 | 0.447869 | 0.233453 | 0.231906 | |
| maxamnt_loans30 | 0.000248 | -0.002764 | -0.000028 | 0.000025 | -0.000864 | -0.001411 | |
| medianamnt_loans30 | 0.044589 | 0.004664 | -0.011610 | -0.005591 | -0.016482 | -0.009467 | |
| cnt_loans90 | 0.004733 | -0.000611 | 0.008962 | 0.009446 | 0.004012 | 0.005141 | |
| amnt_loans90 | 0.199788 | -0.002319 | 0.563496 | 0.567204 | 0.298943 | 0.327436 | |
| maxamnt_loans90 | 0.084144 | -0.001191 | 0.400199 | 0.397251 | 0.234211 | 0.251029 | |
| medianamnt_loans90 | 0.035747 | 0.002771 | -0.037305 | -0.034686 | -0.035489 | -0.034122 | |
| payback30 | 0.048336 | 0.001940 | 0.026915 | 0.019400 | 0.072974 | 0.067110 | |
| payback90 | 0.049183 | 0.002203 | 0.047175 | 0.040800 | 0.095147 | 0.099501 | |

33 rows × 33 columns



```
In [ ]: sns.pairplot(df)
```

```
In [12]: df.skew()
```

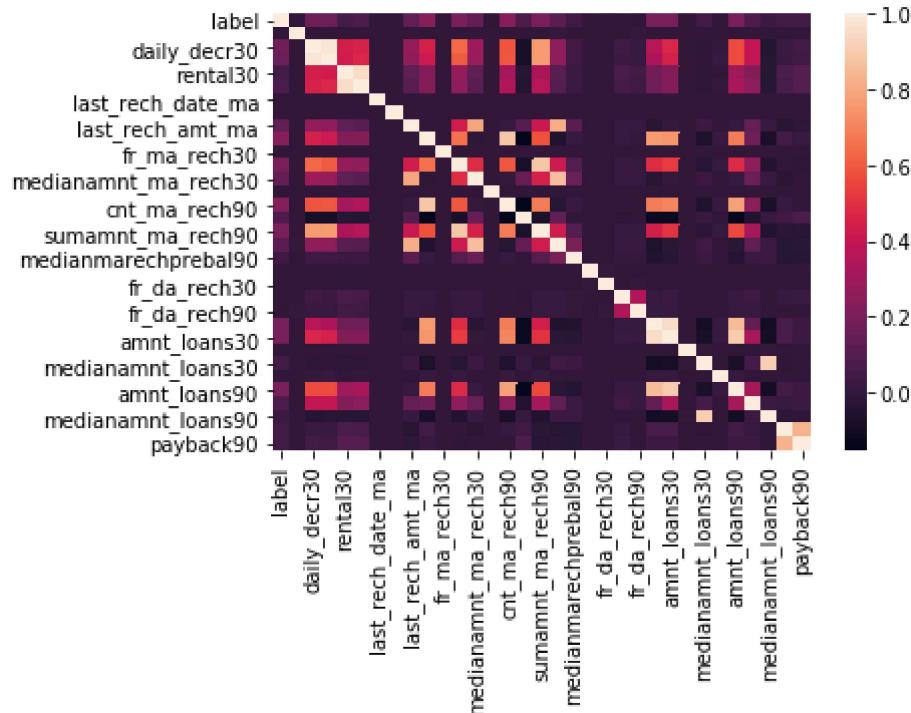
```
Out[12]:
```

| | |
|----------------------|-----------|
| label | -2.270254 |
| aon | 10.392949 |
| daily_decr30 | 3.946230 |
| daily_decr90 | 4.252565 |
| rental30 | 4.521929 |
| rental90 | 4.437681 |
| last_rech_date_ma | 14.790974 |
| last_rech_date_da | 14.814857 |
| last_rech_amt_ma | 3.781149 |
| cnt_ma_rech30 | 3.283842 |
| fr_ma_rech30 | 14.772833 |
| sumamnt_ma_rech30 | 6.386787 |
| medianamnt_ma_rech30 | 3.512324 |
| medianmarechprebal30 | 14.779875 |
| cnt_ma_rech90 | 3.425254 |
| fr_ma_rech90 | 2.285423 |
| sumamnt_ma_rech90 | 4.897950 |
| medianamnt_ma_rech90 | 3.752706 |
| medianmarechprebal90 | 44.880503 |
| cnt_da_rech30 | 17.818364 |
| fr_da_rech30 | 14.776430 |
| cnt_da_rech90 | 27.267278 |
| fr_da_rech90 | 28.988083 |
| cnt_loans30 | 2.713421 |
| amnt_loans30 | 2.975719 |
| maxamnt_loans30 | 17.658052 |
| medianamnt_loans30 | 4.551043 |
| cnt_loans90 | 16.594408 |
| amnt_loans90 | 3.150006 |
| maxamnt_loans90 | 1.678304 |
| medianamnt_loans90 | 4.895720 |
| payback30 | 8.310695 |
| payback90 | 6.899951 |

dtype: float64

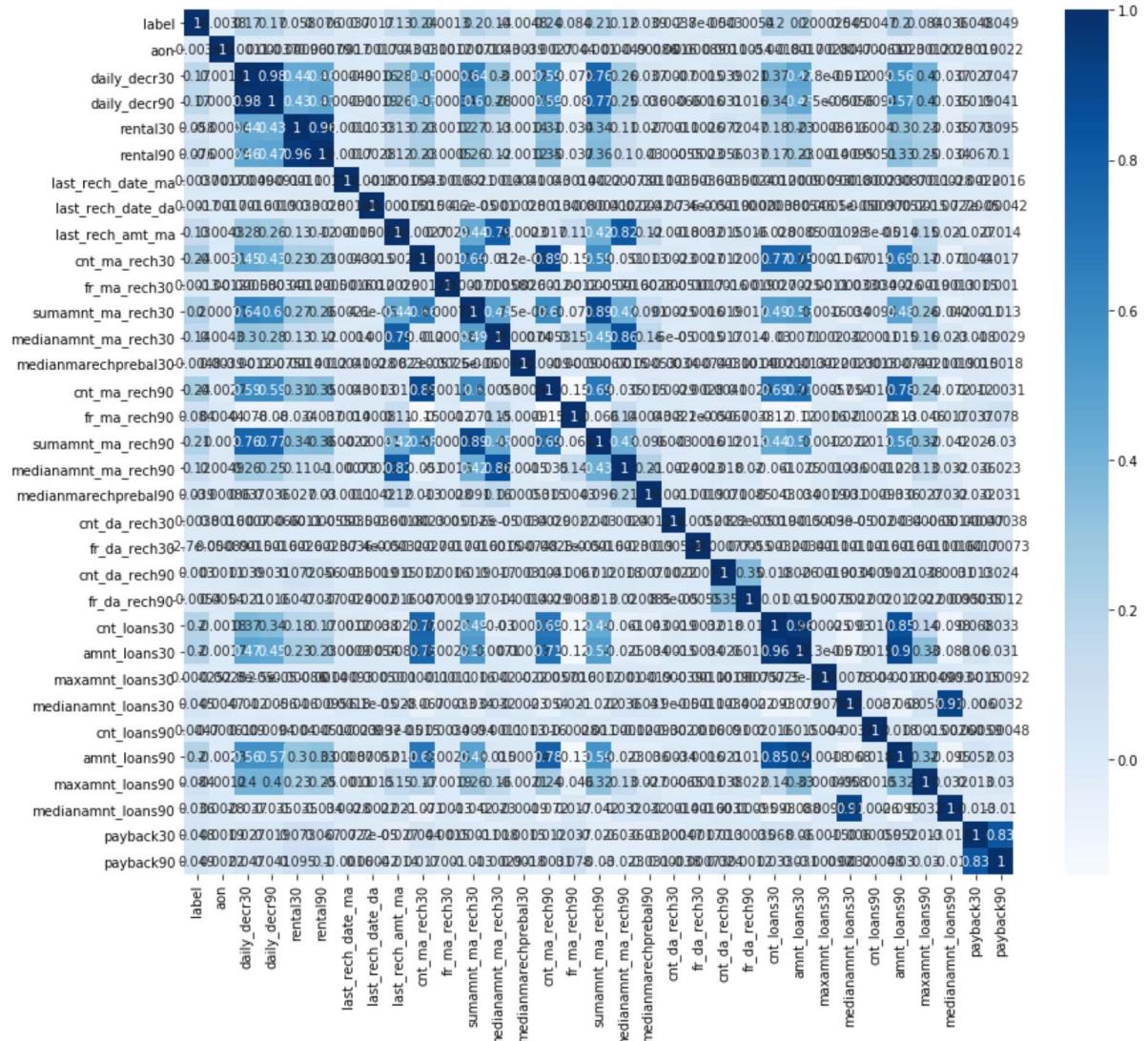
```
In [13]: sns.heatmap(dfcor)
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1f14bcf7208>
```



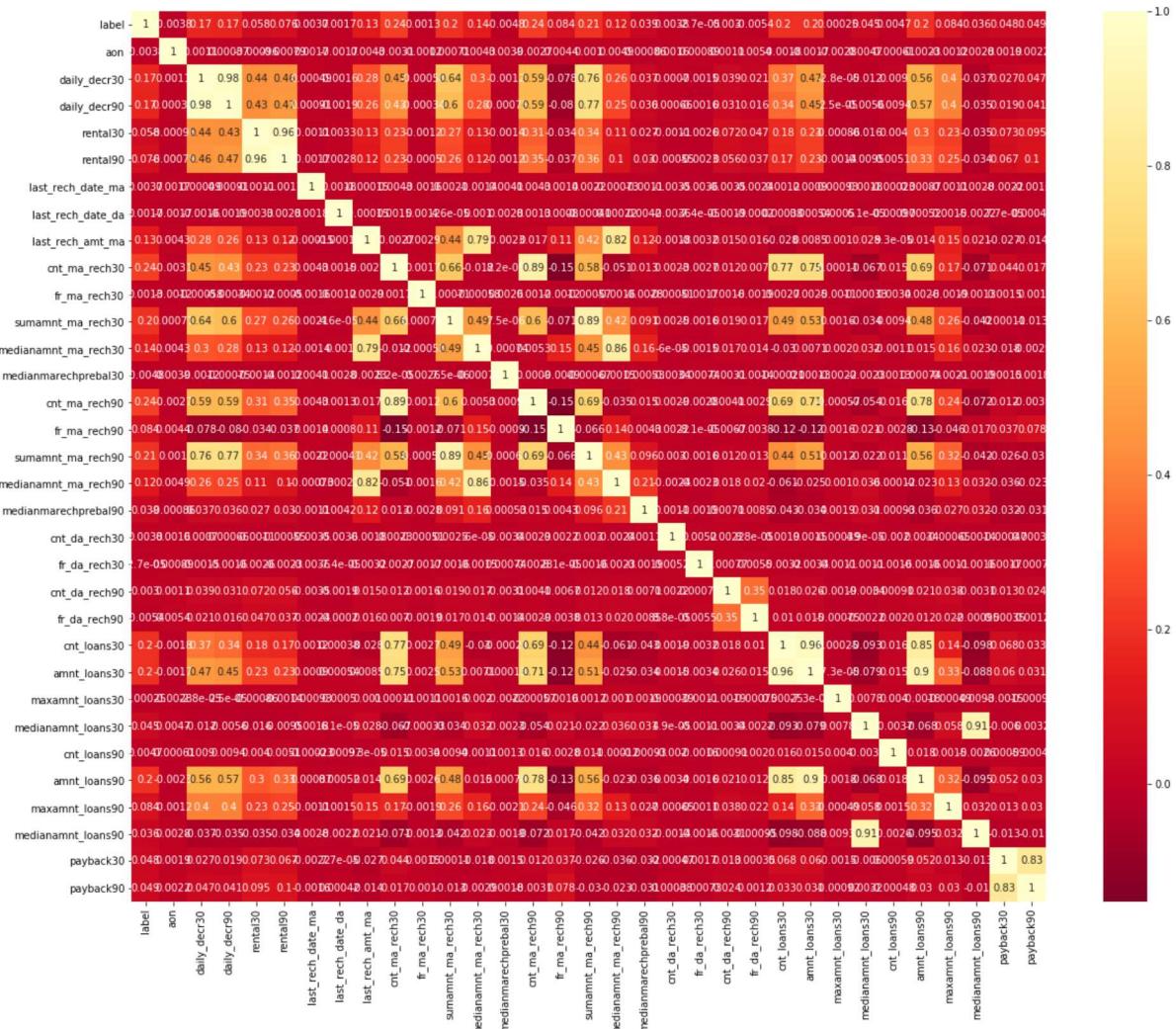
```
In [14]: plt.figure(figsize=(14,12))
sns.heatmap(dfcor,cmap='Blues',annot=True)
```

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1f14c3e8388>



```
In [15]: plt.figure(figsize=(20,16))
sns.heatmap(dfcor, cmap='YlOrRd_r', annot=True)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1f14a60e2c8>



```
In [16]: X=df.iloc[:,1:]
X.head()
```

Out[16]:

| | aon | daily_decr30 | daily_decr90 | rental30 | rental90 | last_rech_date_ma | last_rech_date_da | I |
|---|-------|--------------|--------------|----------|----------|-------------------|-------------------|-----|
| 1 | 272.0 | 3055.050000 | 3065.150000 | 220.13 | 260.13 | | 2.0 | 0.0 |
| 2 | 712.0 | 12122.000000 | 12124.750000 | 3691.26 | 3691.26 | | 20.0 | 0.0 |
| 3 | 535.0 | 1398.000000 | 1398.000000 | 900.13 | 900.13 | | 3.0 | 0.0 |
| 4 | 241.0 | 21.228000 | 21.228000 | 159.42 | 159.42 | | 41.0 | 0.0 |
| 5 | 947.0 | 150.619333 | 150.619333 | 1098.90 | 1098.90 | | 4.0 | 0.0 |

5 rows × 32 columns

```
In [17]: from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x = sc.fit_transform(df)
```

```
In [18]: y=df.iloc[:,0:1]
y.head()
```

Out[18]:

| | label |
|---|-------|
| 1 | 0 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |

```
In [19]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=22,random_state=4
3)
```

```
In [20]: from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

```
In [21]: svc=SVC(kernel='rbf')      # kernel='rbf' ----->default
svc.fit(x_train,y_train)
svc.score(x_train,y_train)
predsvc=svc.predict(x_test)
print(accuracy_score(y_test,predsvc))
print(confusion_matrix(y_test,predsvc))
print(classification_report(y_test,predsvc))
```

| | | | | |
|--------------|-----------|--------|----------|---------|
| 1.0 | | | | |
| [[3 0] | | | | |
| [0 19]] | | | | |
| | precision | recall | f1-score | support |
| | 0 | 1.00 | 1.00 | 1.00 |
| | 1 | 1.00 | 1.00 | 1.00 |
| accuracy | | | 1.00 | 22 |
| macro avg | 1.00 | 1.00 | 1.00 | 22 |
| weighted avg | 1.00 | 1.00 | 1.00 | 22 |

```
In [22]: gnb=GaussianNB()
gnb.fit(x_train,y_train)
gnb.score(x_train,y_train)
predsvc=gnb.predict(x_test)
print(accuracy_score(y_test,predsvc))
print(confusion_matrix(y_test,predsvc))
print(classification_report(y_test,predsvc))
```

```
1.0
[[ 3  0]
 [ 0 19]]
      precision    recall   f1-score   support
          0       1.00     1.00     1.00      3
          1       1.00     1.00     1.00     19

   accuracy                           1.00      22
  macro avg       1.00     1.00     1.00      22
weighted avg       1.00     1.00     1.00      22
```

```
In [23]: #DecisionTreeClassifier(criterion='gini')----->default
#DecisionTreeClassifier(criterion='entropy')
#Gini and entropy
dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)
dtc.score(x_train,y_train)
preddtc=dtc.predict(x_test)
print(accuracy_score(y_test,preddtc))
print(confusion_matrix(y_test,preddtc))
print(classification_report(y_test,preddtc))
```

```
1.0
[[ 3  0]
 [ 0 19]]
      precision    recall   f1-score   support
          0       1.00     1.00     1.00      3
          1       1.00     1.00     1.00     19

   accuracy                           1.00      22
  macro avg       1.00     1.00     1.00      22
weighted avg       1.00     1.00     1.00      22
```

In [24]: #k=10

```

knn=KNeighborsClassifier()
knn.fit(x_train, y_train)
knn.score(x_train,y_train)
predknn=dtc.predict(x_test)
print(accuracy_score(y_test,preddtc))
print(confusion_matrix(y_test,preddtc))
print(classification_report(y_test,preddtc))

```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 1.00 | 1.00 | 1.00 | 3 |
| 1 | 1.00 | 1.00 | 1.00 | 19 |
| accuracy | | | 1.00 | 22 |
| macro avg | 1.00 | 1.00 | 1.00 | 22 |
| weighted avg | 1.00 | 1.00 | 1.00 | 22 |

In [25]: for k in range(25):

```

k_value=k+1
knn=KNeighborsClassifier(n_neighbors=k_value)
knn.fit(x_train, y_train)
y_pred=knn.predict(x_test)
print("accuracy score is ", accuracy_score(y_test,y_pred), 'at k-value',k_value)

```

accuracy score is 1.0 at k-value 1
 accuracy score is 1.0 at k-value 2
 accuracy score is 1.0 at k-value 3
 accuracy score is 1.0 at k-value 4
 accuracy score is 1.0 at k-value 5
 accuracy score is 1.0 at k-value 6
 accuracy score is 1.0 at k-value 7
 accuracy score is 1.0 at k-value 8
 accuracy score is 1.0 at k-value 9
 accuracy score is 1.0 at k-value 10
 accuracy score is 1.0 at k-value 11
 accuracy score is 1.0 at k-value 12
 accuracy score is 1.0 at k-value 13
 accuracy score is 1.0 at k-value 14
 accuracy score is 1.0 at k-value 15
 accuracy score is 1.0 at k-value 16
 accuracy score is 1.0 at k-value 17
 accuracy score is 1.0 at k-value 18
 accuracy score is 1.0 at k-value 19
 accuracy score is 1.0 at k-value 20
 accuracy score is 1.0 at k-value 21
 accuracy score is 1.0 at k-value 22
 accuracy score is 1.0 at k-value 23
 accuracy score is 1.0 at k-value 24
 accuracy score is 1.0 at k-value 25

```
In [26]: svc=SVC(kernel='poly')
svc.fit(x_train,y_train)
svc.score(x_train,y_train)
predsvc=svc.predict(x_test)
print(accuracy_score(y_test,predsvc))
print(confusion_matrix(y_test,predsvc))
print(classification_report(y_test,predsvc))
```

```
1.0
[[ 3  0]
 [ 0 19]]
      precision    recall   f1-score   support
          0       1.00     1.00     1.00        3
          1       1.00     1.00     1.00       19

      accuracy                           1.00        22
     macro avg       1.00     1.00     1.00        22
weighted avg       1.00     1.00     1.00        22
```

```
In [27]: def svmkernel(ker):
    svc=SVC(kernel=ker)
    svc.fit(x_train,y_train)
    svc.score(x_train,y_train)
    predsvc=svc.predict(x_test)
    print(accuracy_score(y_test,predsvc))
    print(confusion_matrix(y_test,predsvc))
    print(classification_report(y_test,predsvc))
```

```
In [28]: svmkernel('rbf')
```

```
1.0
[[ 3  0]
 [ 0 19]]
      precision    recall   f1-score   support
          0       1.00     1.00     1.00        3
          1       1.00     1.00     1.00       19

      accuracy                           1.00        22
     macro avg       1.00     1.00     1.00        22
weighted avg       1.00     1.00     1.00        22
```

```
In [29]: svmkernel('poly')
```

```
1.0
[[ 3  0]
 [ 0 19]]
      precision    recall   f1-score   support
          0         1.00     1.00     1.00        3
          1         1.00     1.00     1.00       19
accuracy                           1.00        22
macro avg       1.00     1.00     1.00        22
weighted avg    1.00     1.00     1.00        22
```

```
In [30]: from sklearn.model_selection import cross_val_score
svscores =cross_val_score(svc,x,y,cv=5)
print(svscores)
print(svscores.mean(),svscores.std())
```

```
[0.99997614 0.99992843 0.99995229 0.99992843 0.99995229]
0.9999475172629093 1.7852148893725777e-05
```

```
In [31]: gnbscores = cross_val_score(gnb, x,y, cv=5)
print(gnbscores)
print(gnbscores.mean(),gnbscores.std())
```

```
[1. 1. 1. 1. 1.]
1.0 0.0
```

```
In [32]: from sklearn import svm, datasets
from sklearn.model_selection import GridSearchCV
parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}
svc = svm.SVC()
clf = GridSearchCV(svc, parameters)
clf.fit(x_test, y_test)
clf.fit(x_train, y_train)

sorted(clf.cv_results_.keys())
```

```
Out[32]: ['mean_fit_time',
'mean_score_time',
'mean_test_score',
'param_C',
'param_kernel',
'params',
'rank_test_score',
'split0_test_score',
'split1_test_score',
'split2_test_score',
'split3_test_score',
'split4_test_score',
'std_fit_time',
'std_score_time',
'std_test_score']
```

```
In [33]: import pickle
filename = 'picklegnbfile.pkl'
pickle.dump(dtc, open(filename, 'wb'))

#Load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))

loaded_model.predict(x_test)
```

```
Out[33]: array([1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],  
dtype=int64)
```