```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/c
```

```
import pandas as pd
import numpy as np
import re

import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.metrics import accuracy_score, log_loss
from sklearn.model_selection import StratifiedKFold

import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM, Dropout, Bidirectional
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.utils import plot_model, to_categorical
from tensorflow.keras.optimizers import Adam

from keras.utils import np_utils

import warnings
warnings.filterwarnings(action='ignore')
train = pd.read_csv("/content/drive/MyDrive/data/credit/train.csv")
duptrain = pd.read_csv("/content/drive/MyDrive/data/credit/train.csv")
test = pd.read_csv("/content/drive/MyDrive/data/credit/test.csv")
sample_submission = pd.read_csv("/content/drive/MyDrive/data/novel/sample_submission.csv")
```

```
train.shape
```

```
(26457, 20)
```

중복값(약 1600)제거후26457남음 중복값 기준 인덱스를 제외한 나머지 칼럼들

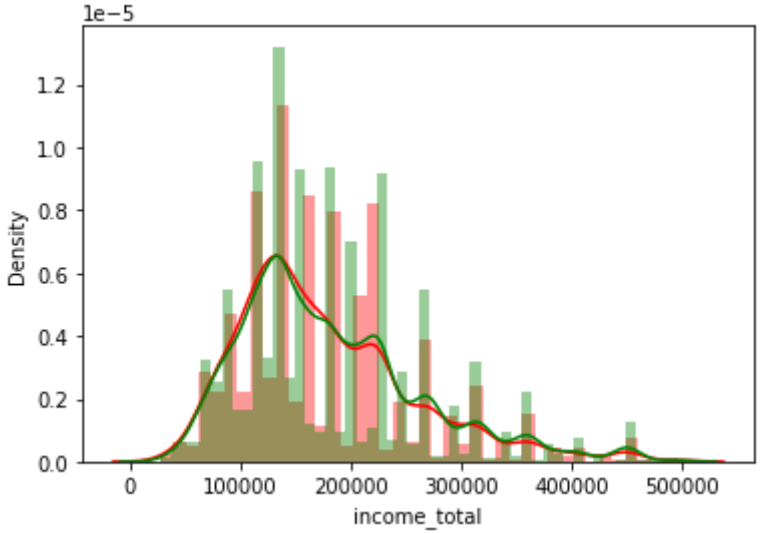
```
train.describe()
```

	index	child_num	income_total	DAYS_BIRTH	DAYS_EMPLOYED	FLAG_M(
count	26457.000000	26457.000000	2.645700e+04	26457.000000	26457.000000	264
mean	13228.000000	0.428658	1.873065e+05	-15958.053899	59068.750728	
std	7637.622372	0.747326	1.018784e+05	4201.589022	137475.427503	
min	0.000000	0.000000	2.700000e+04	-25152.000000	-15713.000000	

```
train['YEAR_BIRTH']=train['DAYS_BIRTH']//-365
train['YEAR_begin']=train['begin_month']//-12
```

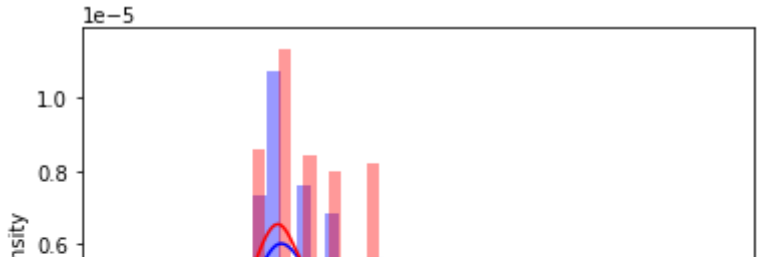
```
cond2 = (train['credit'] == 2)
cond1 = (train['credit'] == 1)
cond_amt = (train['income_total'] < 500000)
sns.distplot(train[cond1 & cond_amt]['income_total'], label='1', color='red')
sns.distplot(train[cond2 & cond_amt]['income_total'], label='2', color='green')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f00aa868e50>
```

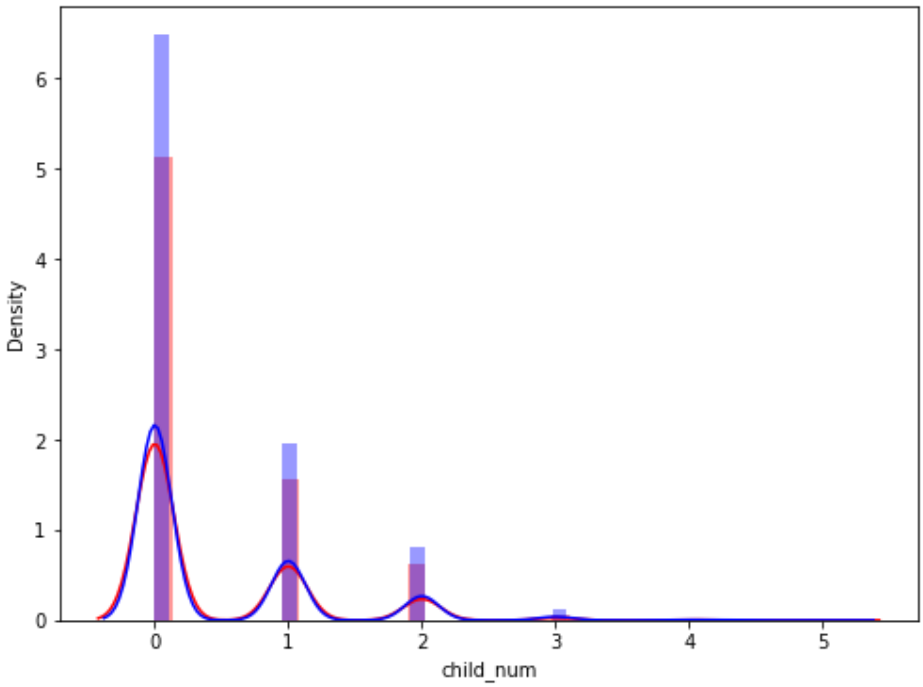


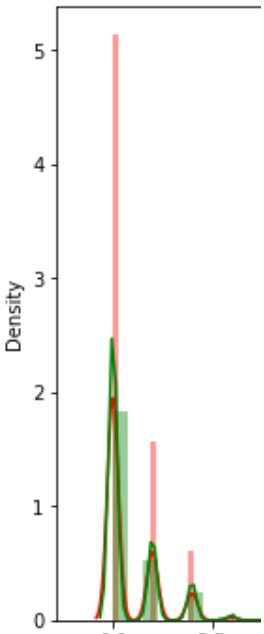

```
cond1 = (train['credit'] == 1)
cond0 = (train['credit'] == 0)

cond_amt = (train['income_total'] < 500000)

sns.distplot(train[cond0 & cond_amt]['income_total'], label='0', color='blue')
sns.distplot(train[cond1 & cond_amt]['income_total'], label='1', color='red')
```

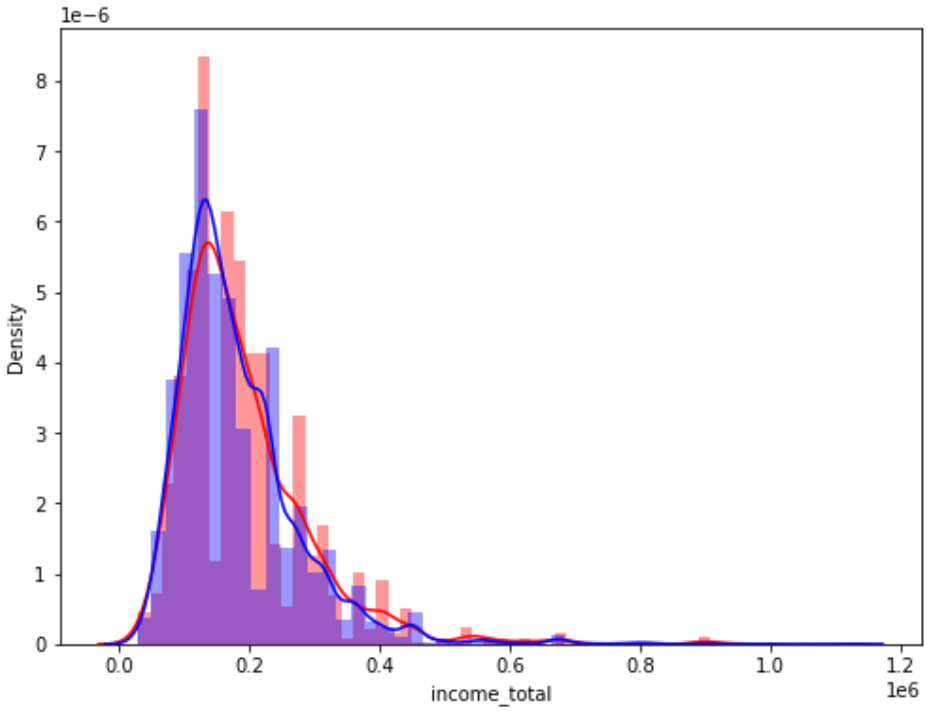
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f00a974f190>
```

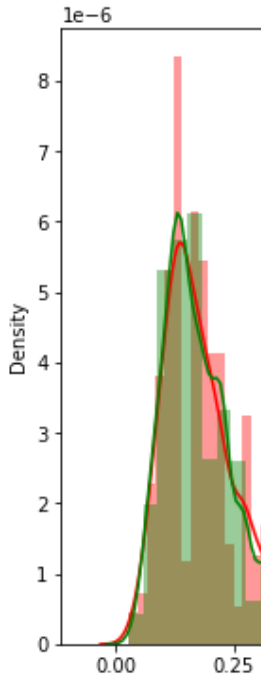


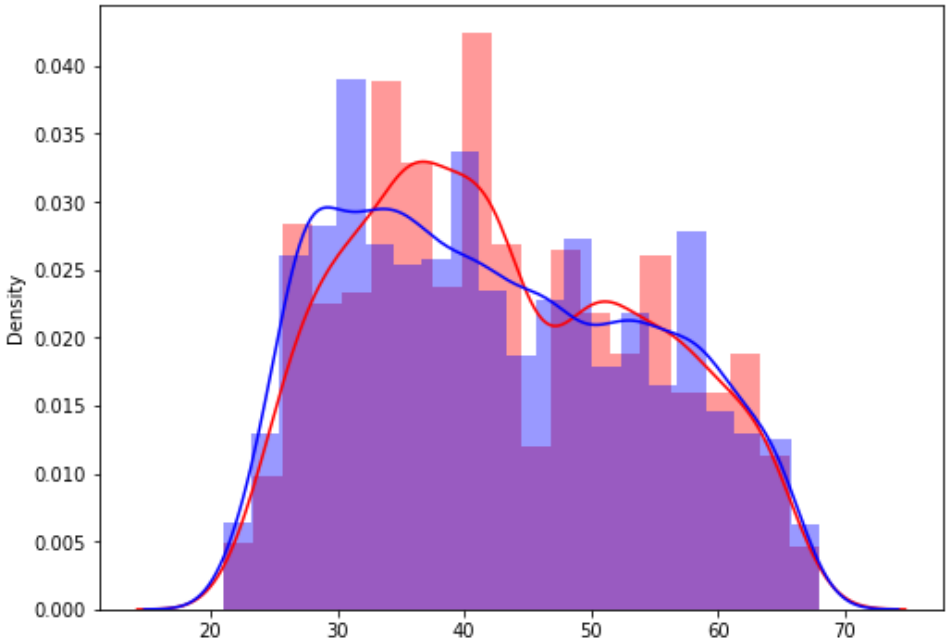

```
def show_hist_by_credit(train, columns):
    cond_2 = (train['credit'] == 2)
    cond_1 = (train['credit'] == 1)
    cond_0 = (train['credit'] == 0)

    for column in columns:
        fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(18, 6), squeeze=False)
        sns.distplot(train[cond_0][column], ax=axs[0][0], label='0', color='red')
        sns.distplot(train[cond_1][column], ax=axs[0][0], label='1', color='blue')
        sns.distplot(train[cond_0][column], ax=axs[0][1], label='0', color='red')
        sns.distplot(train[cond_2][column], ax=axs[0][1], label='2', color='green')


columns = ['child_num', 'income_total', 'YEAR_BIRTH', 'DAYS_EMPLOYED',  'FLAG_MOBIL', 'family_size'

show_hist_by_credit(train, columns)
```

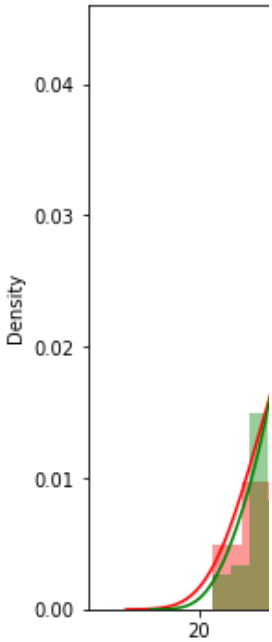

Density
child_num

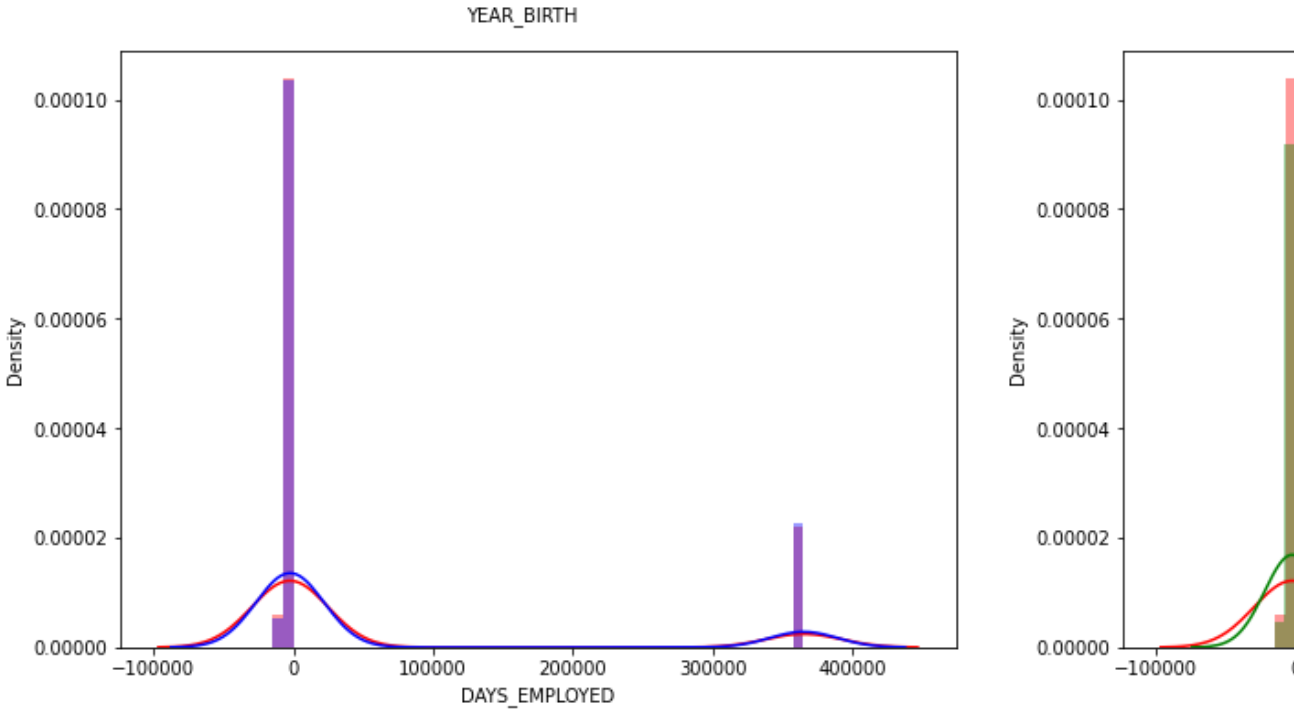

Density

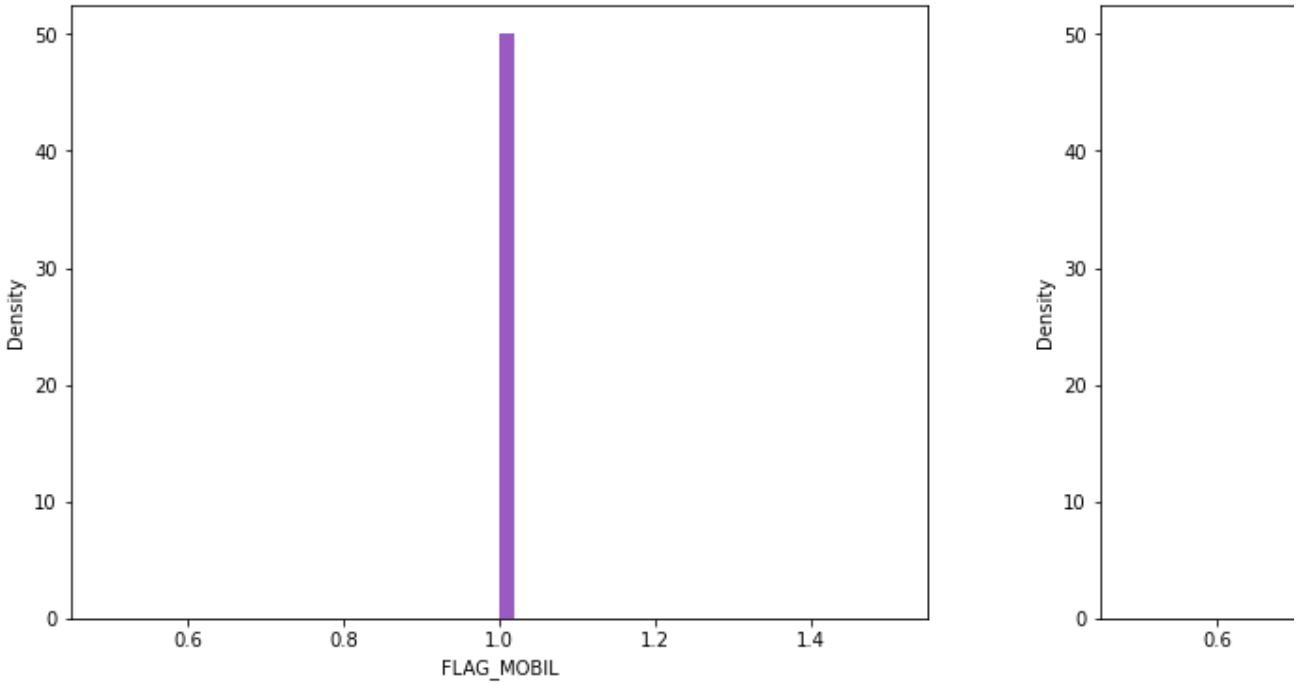

1e−6
Density
income_total
1e6

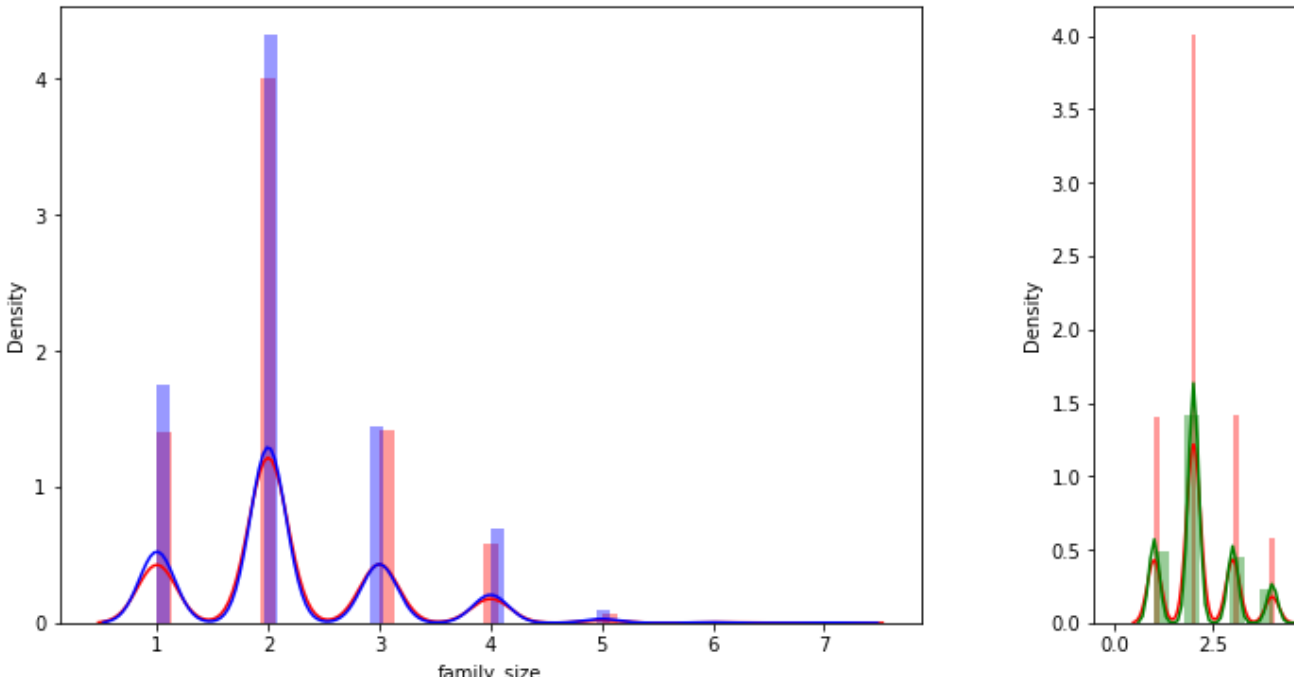

1e−6
Density

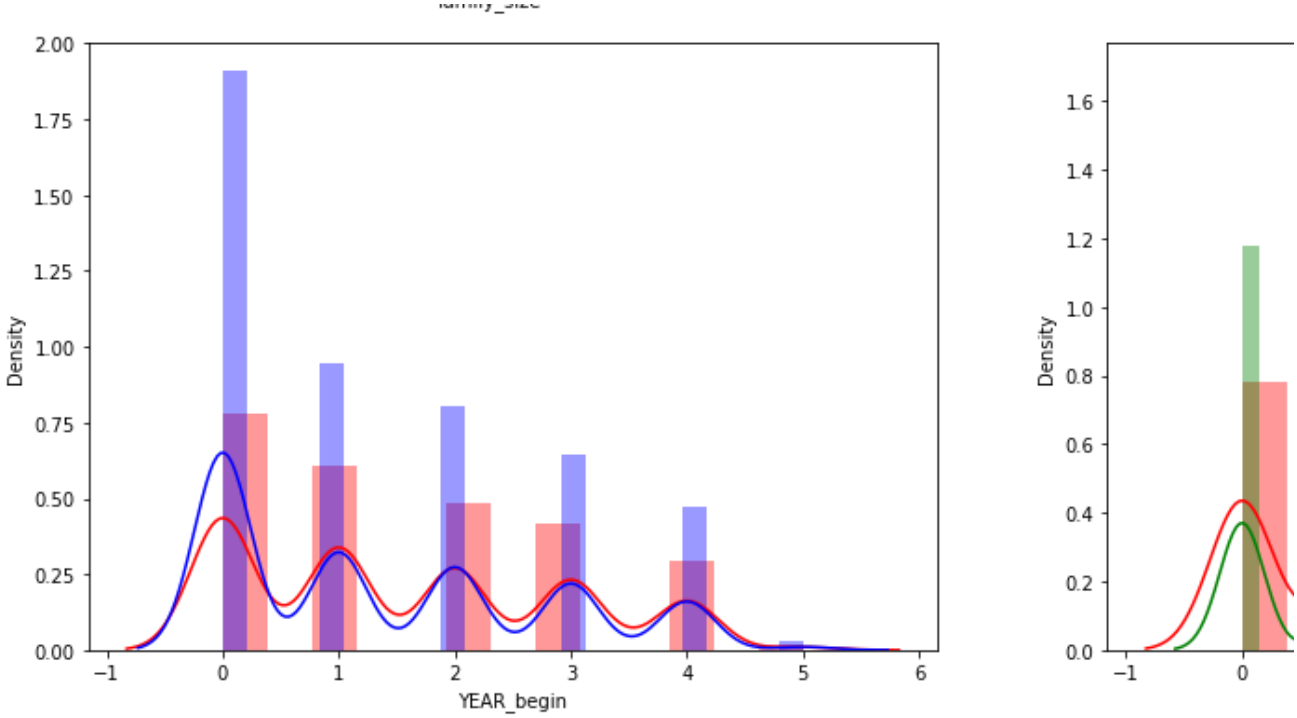

Density


Density


YEAR_BIRTH
Density
DAYS_EMPLOYED


Density
FLAG_MOBIL


Density
family_size



1열은 신용 0과1을 비교하였다. 살아온날을 기점으로 40언저리가 신용이 높고 27언저리가 신용이 낮다는것을 알수있다. 2열은 신용 0과2을 비교하였다. 살아온날을 기점으로 40언저리가 신용도가 낮게 나왔다. 두가지를 공통적으로 보았을 때 확연한 차이는 업무 시작일로 부터 차이가 나는것을 알았다.

따라서 나이에 대해 eda 생성시 카테고리로 미취업자,30대/40대/50대 ,정년(변수생성) 으로 나눌 예정

2. 중복된 값(ex: 동일한 사람이 신용카드 복수 발급)이 약 1600개 있는것을 확인 이또한 중복자를 변수로 생성하여 eda 작성예정 ex: 카드 2개 발급자, 3개 발급자로 나눌 예정(변수 생성)

```
col_range = ['child_num', 'income_total', 'DAYS_BIRTH', 'DAYS_EMPLOYED',  'FLAG_MOBIL', 'work_phone
```

```
duptrain=duptrain.drop('index',axis=1)
```

```
duptrain.drop_duplicates()
```

	gender	car	reality	child_num	income_total	income_type	edu_type	fami
0	F	N	N	0	202500.0	Commercial associate	Higher education	
1	F	N	Y	1	247500.0	Commercial associate	Secondary / secondary special	Civil r
2	M	Y	Y	0	450000.0	Working	Higher education	
3	F	N	Y	0	202500.0	Commercial associate	Secondary / secondary special	
4	F	Y	Y	0	157500.0	State servant	Higher education	
...	...	...	...	...	...	...	...	
26452	F	N	N	2	225000.0	State servant	Secondary / secondary special	
26453	F	N	Y	1	180000.0	Working	Higher education	Se

중복값(약 1600)제거후약 24000남음

							secondary	
26455	M	N	Y	0	171000.0	Working	Incomplete higher	Civil

```
train.head(2)
```

	index	gender	car	reality	child_num	income_total	income_type	edu_type	fa
0	0	F	N	N	0	202500.0	Commercial associate	Higher education	
1	1	F	N	Y	1	247500.0	Commercial associate	Secondary / secondary special	Civ

✓ 0초 오후 4:37에 완료됨