In [106]:
```python
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [107]:
```python
df=pd.read_csv(r'C:\Users\sharma17\Downloads\ml\classification dataset\bank.csv')
```

In [108]:
```python
df.head()
```

Out[108]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	deposit
0	59	0	1	1	0	2343	1	0	2	5	8	1042	1	-1	0	3	1
1	56	0	1	1	0	45	0	0	2	5	8	1467	1	-1	0	3	1
2	41	9	1	1	0	1270	1	0	2	5	8	1389	1	-1	0	3	1
3	55	7	1	1	0	2476	1	0	2	5	8	579	1	-1	0	3	1
4	54	0	1	2	0	184	0	0	2	5	8	673	2	-1	0	3	1

In [109]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11162 entries, 0 to 11161
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        11162 non-null  int64
 1   job        11162 non-null  int64
 2   marital    11162 non-null  int64
 3   education  11162 non-null  int64
 4   default    11162 non-null  int64
 5   balance    11162 non-null  int64
 6   housing    11162 non-null  int64
 7   loan       11162 non-null  int64
 8   contact    11162 non-null  int64
 9   day        11162 non-null  int64
 10  month      11162 non-null  int64
 11  duration   11162 non-null  int64
 12  campaign   11162 non-null  int64
 13  pdays      11162 non-null  int64
 14  previous   11162 non-null  int64
 15  poutcome   11162 non-null  int64
 16  deposit    11162 non-null  int64
dtypes: int64(17)
memory usage: 1.4 MB
```

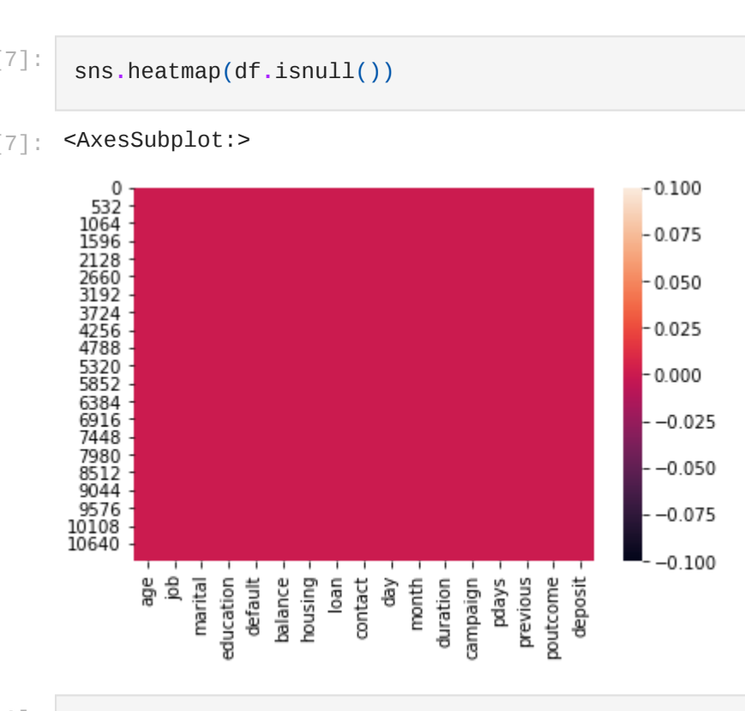
In [110]:
```python
df.describe()
```

Out[110]:

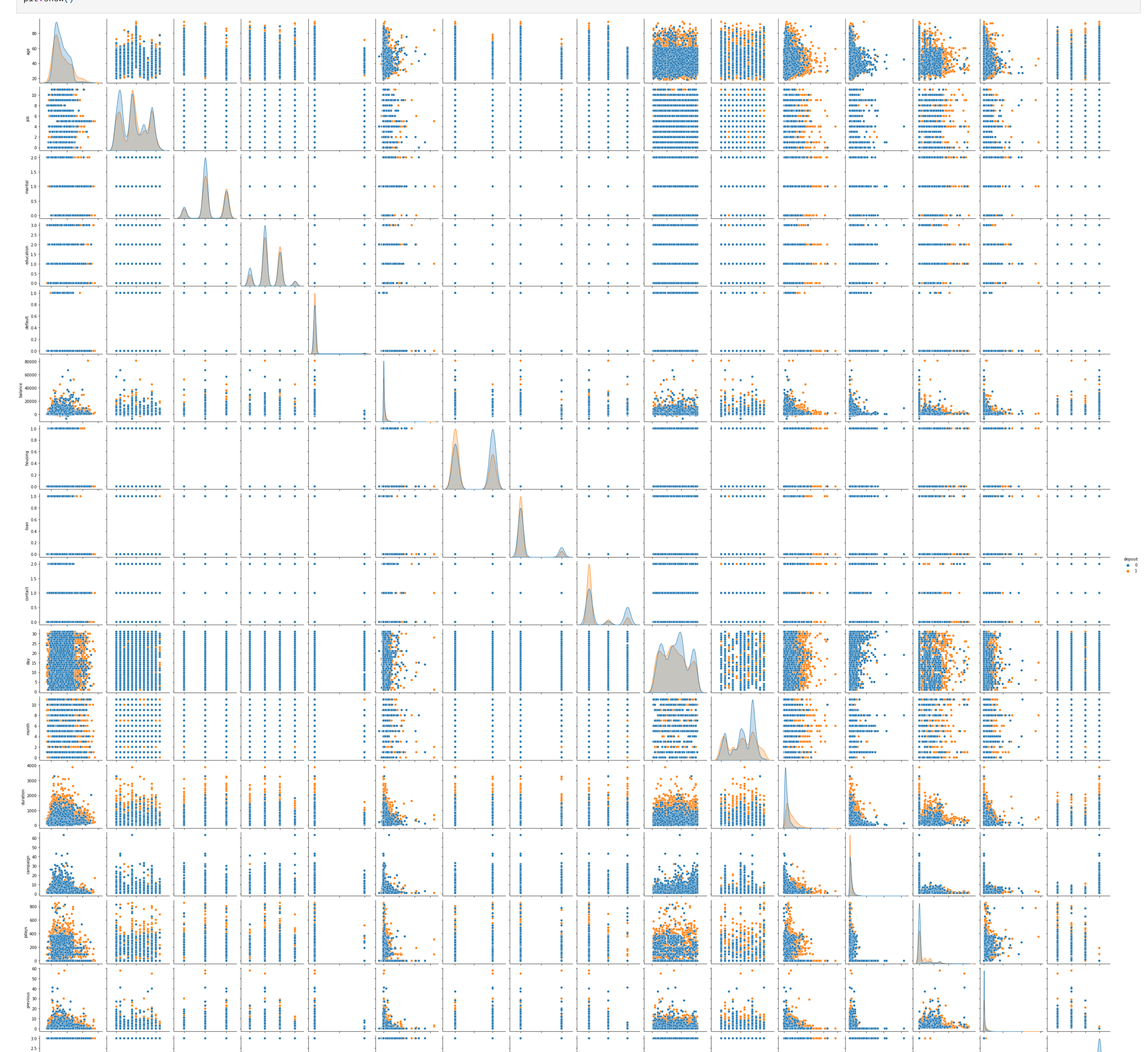
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	deposit
count	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000
mean	41.231948	4.487905	1.199337	1.285164	0.015051	1528.538524	0.473123	0.130801	0.489697	15.658036	5.445709	371.993818	2.508421	51.330407	0.832557	2.477782	0.473840
std	11.913369	3.225132	0.625552	0.749478	0.121761	3225.413326	0.499299	0.337198	0.818724	8.420740	3.191939	347.128386	2.722077	108.758282	2.292007	1.002952	0.499338
min	18.000000	0.000000	0.000000	0.000000	0.000000	-6847.000000	0.000000	0.000000	0.000000	1.000000	0.000000	2.000000	1.000000	-1.000000	0.000000	0.000000	0.000000
25%	32.000000	1.000000	1.000000	1.000000	0.000000	122.000000	0.000000	0.000000	0.000000	8.000000	3.000000	138.000000	1.000000	-1.000000	0.000000	2.000000	0.000000
50%	39.000000	4.000000	1.000000	1.000000	0.000000	550.000000	0.000000	0.000000	0.000000	15.000000	6.000000	255.000000	2.000000	-1.000000	0.000000	3.000000	0.000000
75%	49.000000	7.000000	2.000000	2.000000	0.000000	1708.000000	1.000000	0.000000	2.000000	22.000000	8.000000	496.000000	3.000000	20.750000	1.000000	3.000000	1.000000
max	95.000000	11.000000	2.000000	3.000000	1.000000	81204.000000	1.000000	1.000000	2.000000	31.000000	11.000000	3881.000000	63.000000	854.000000	58.000000	3.000000	1.000000

In [7]:
```python
sns.heatmap(df.isnull())
```

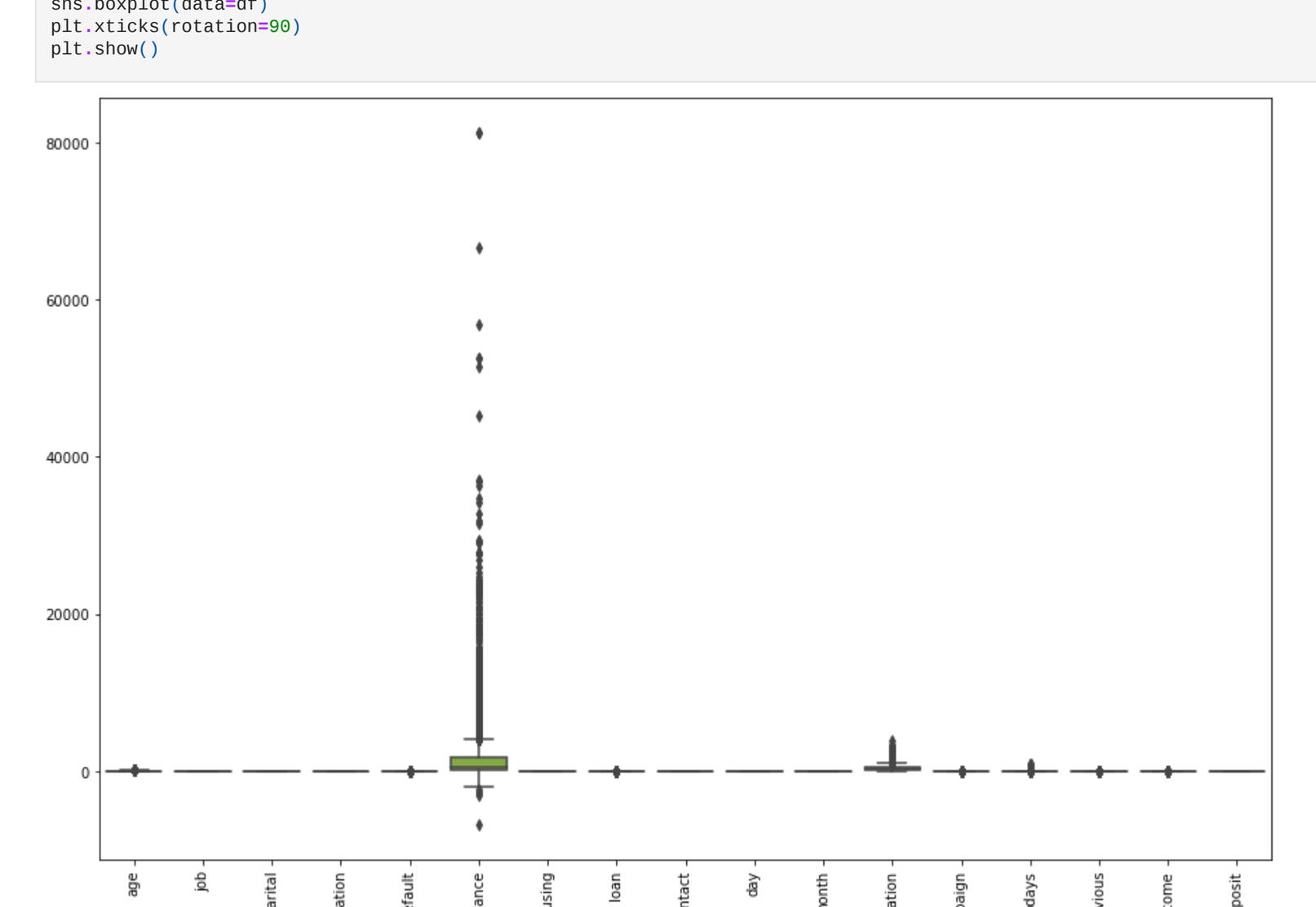
Out[7]: <AxesSubplot:>



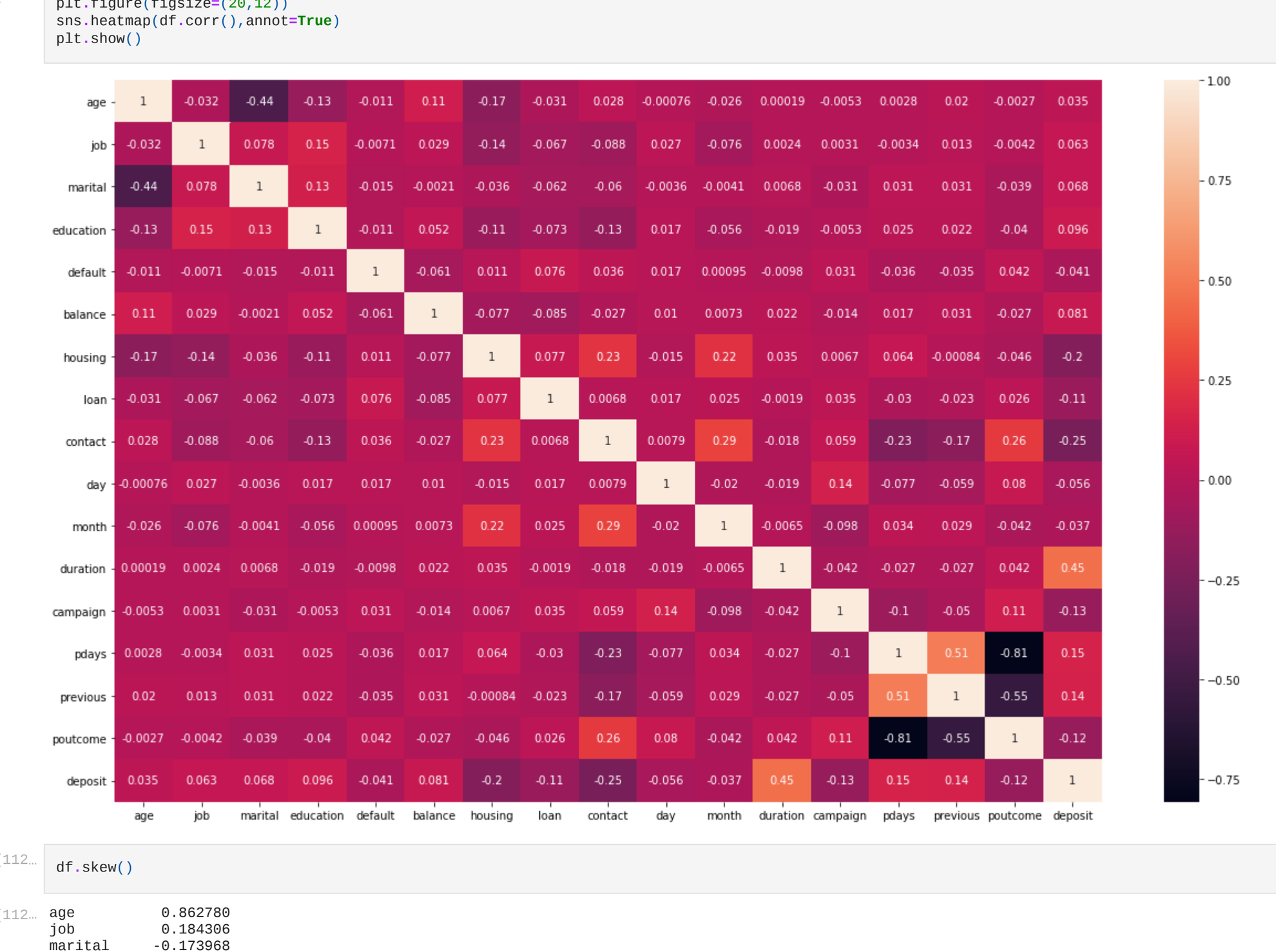
In [32]:
```python
sns.pairplot(data=df,hue='deposit')
plt.show()
```



In [21]:
```python
plt.figure(figsize=(15,10))
sns.boxplot(data=df)
plt.xticks(rotation=90)
plt.show()
```



In [111]:
```python
plt.figure(figsize=(20,12))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



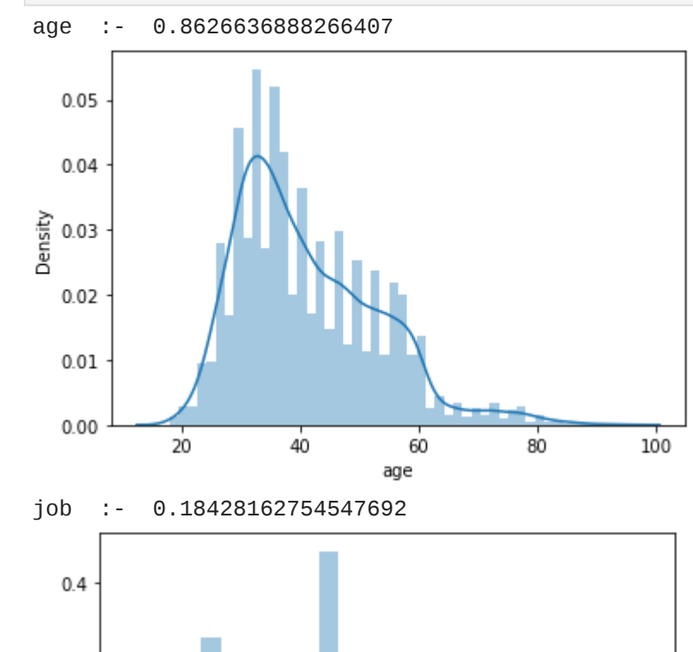
In [112]:
```python
df.skew()
```

Out[112]:
```
age          0.862780
job          0.184306
marital     -0.173968
education    0.115543
default      7.966983
balance      8.224619
housing      0.107678
loan         2.190201
contact      1.182309
day          0.111330
month       -0.322998
duration     2.143695
campaign     5.545578
pdays        2.449986
previous     7.335298
poutcome    -1.718276
deposit      0.104798
dtype: float64
```

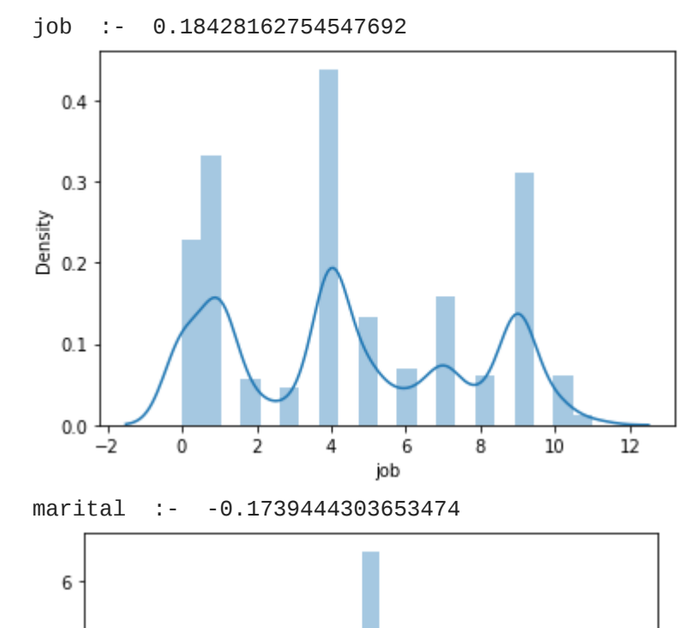
In [113]:
```python
from scipy.stats import skew
```

In [114]:
```python
for col in df:

    print(col,  ' :-  ',end='')
    print(skew(df[col]))


    sns.distplot(df[col])
    plt.show()
```

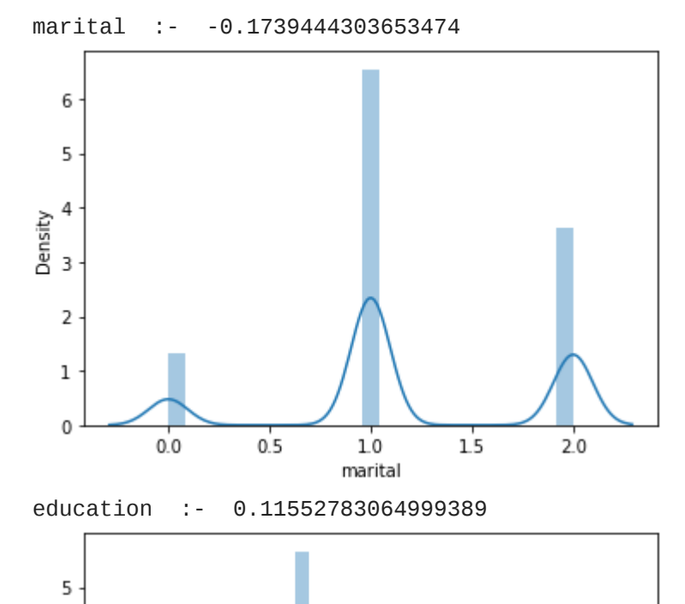
age :- 0.8626636888266407



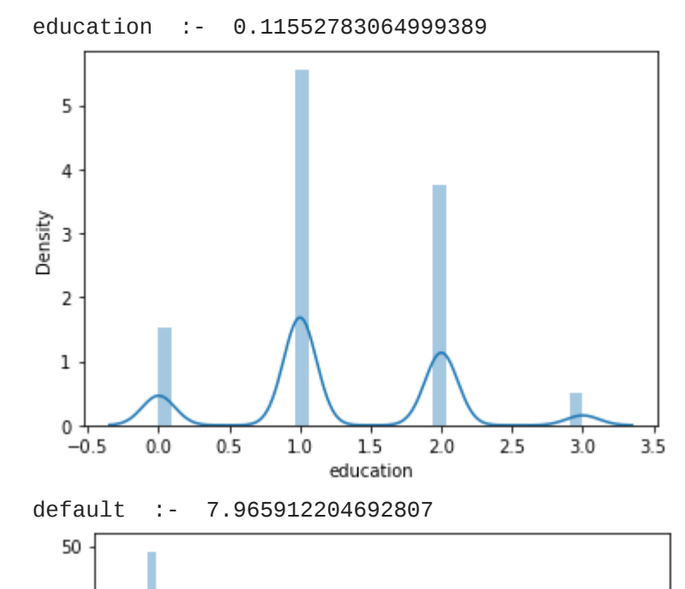
job :- 0.18428162754547692



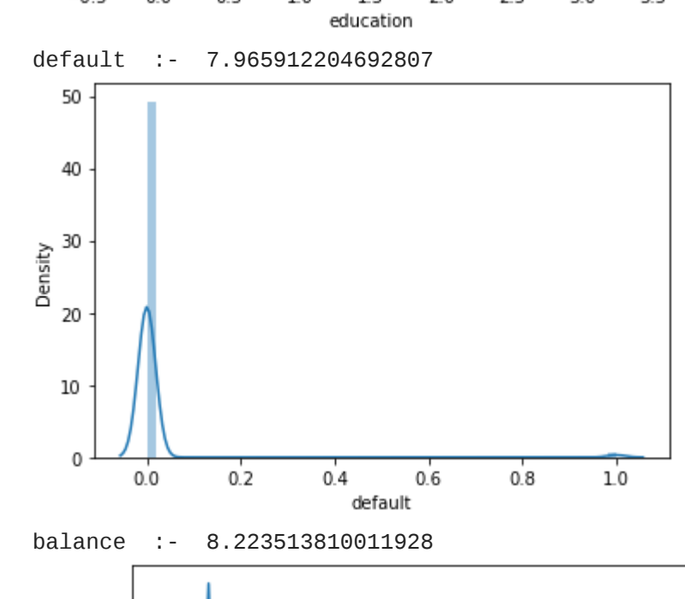
marital :- -0.1739444303653474



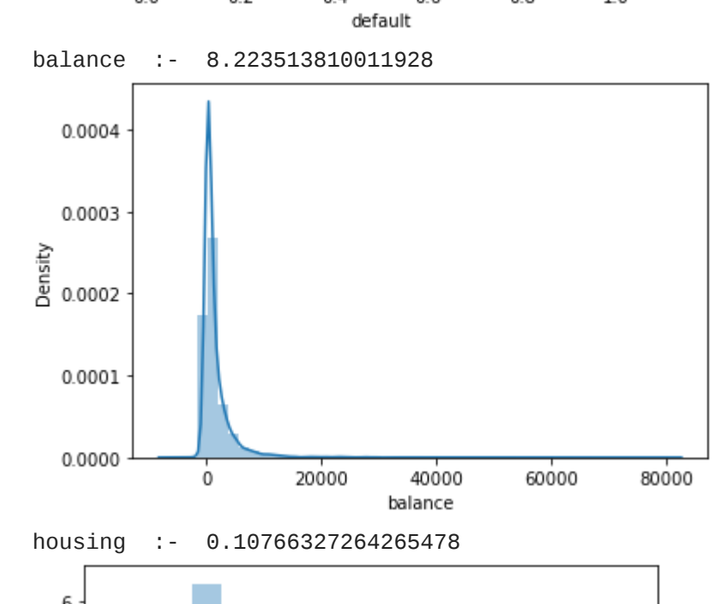
education :- 0.11552783064999389



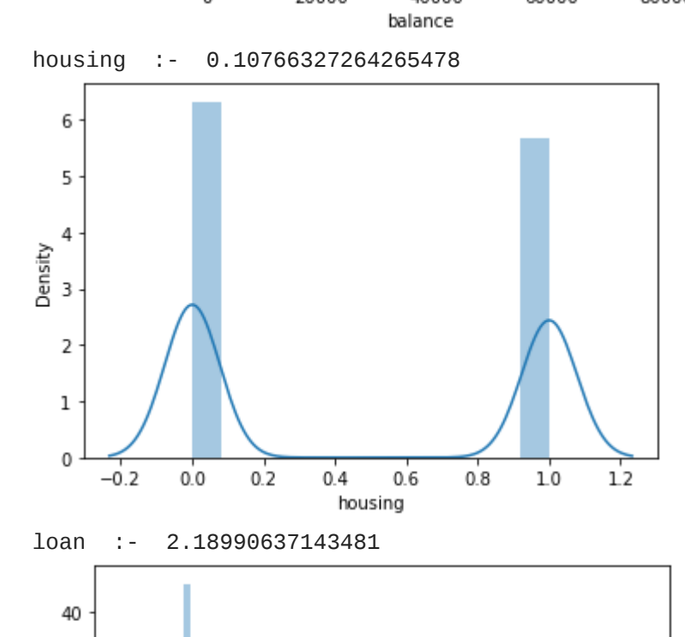
default :- 7.965912204692807



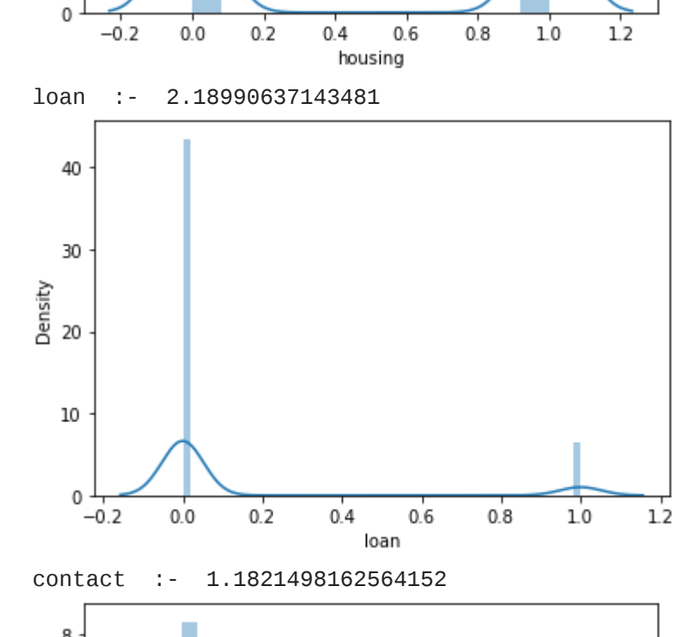
balance :- 8.223513810011928



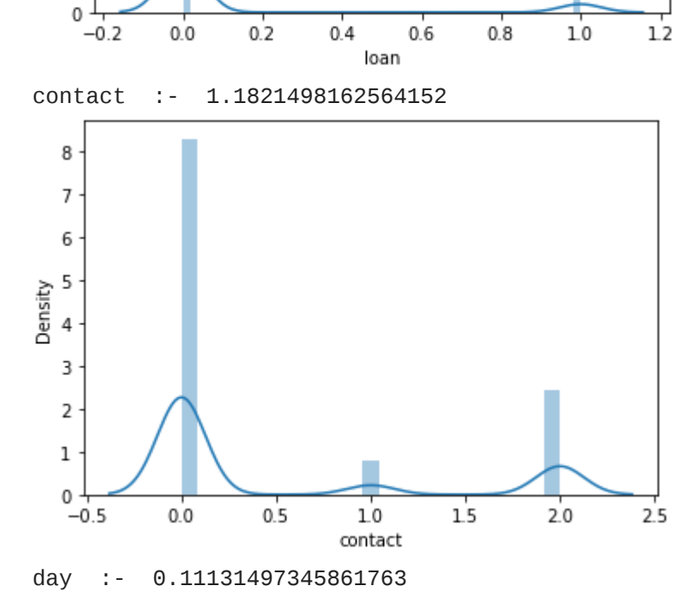
housing :- 0.10766327264265478



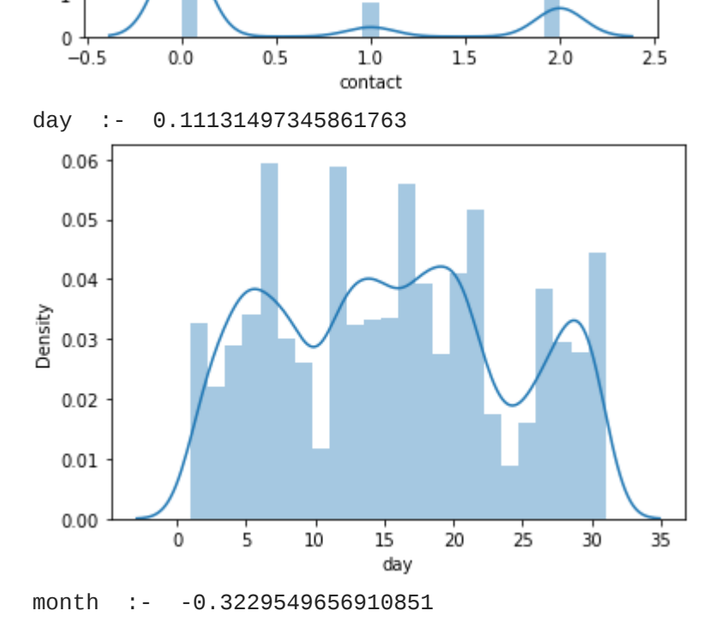
loan :- 2.18990637143481



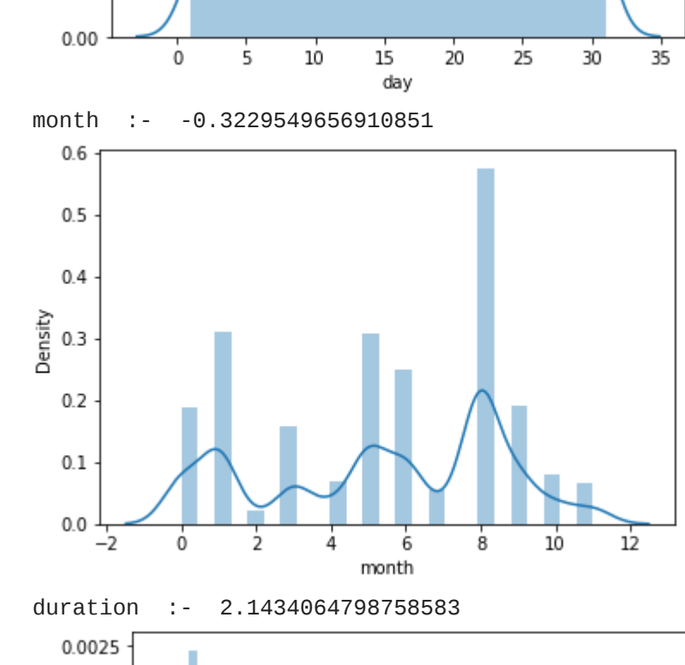
contact :- 1.1821498162564152



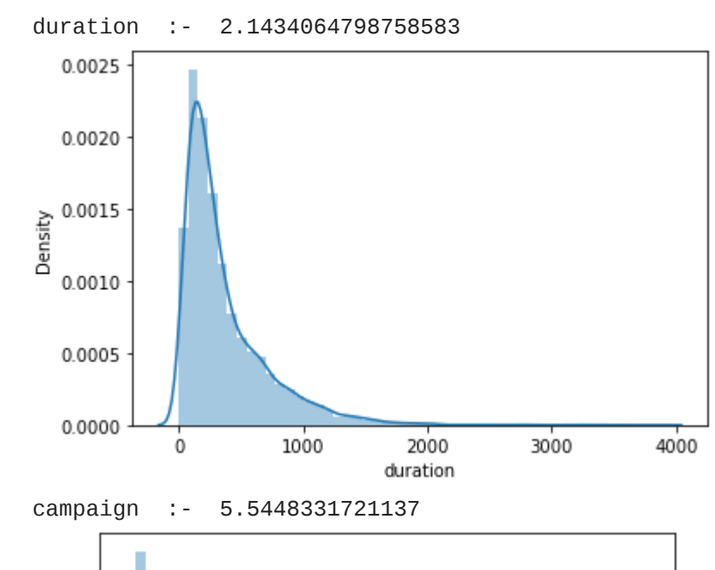
day :- 0.11131497345861763



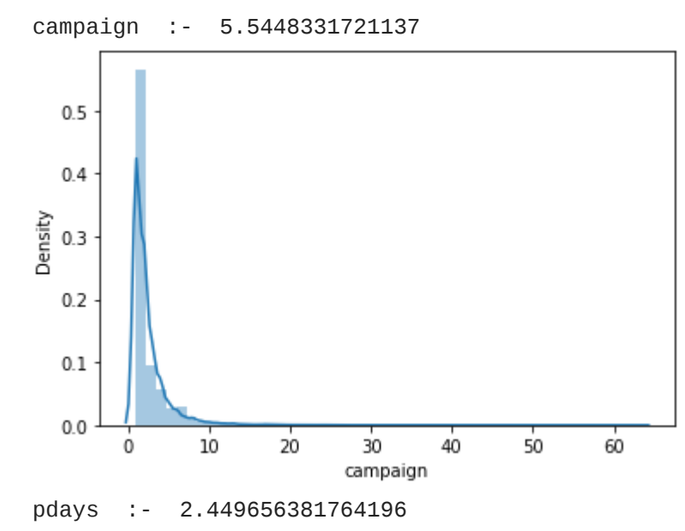
month :- -0.3229549656910851



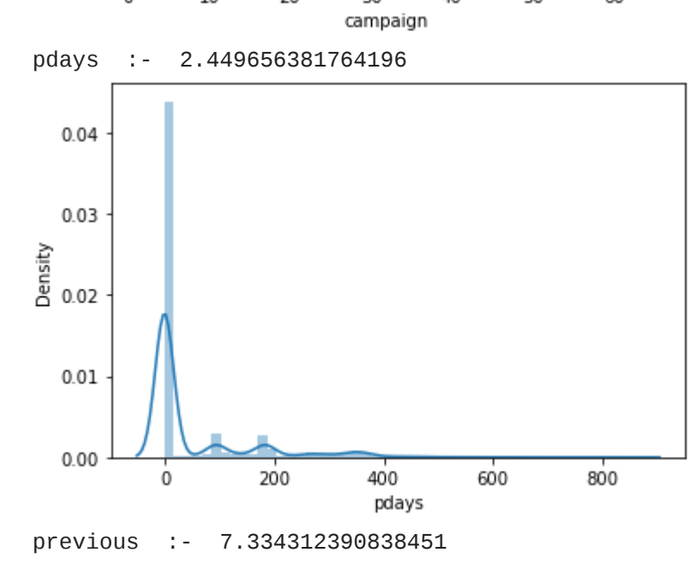
duration :- 2.1434064798758583



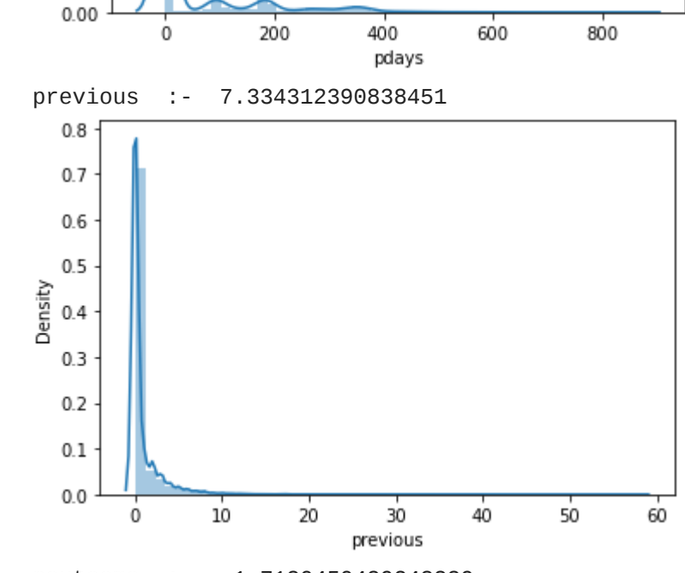
campaign :- 5.5448331721137



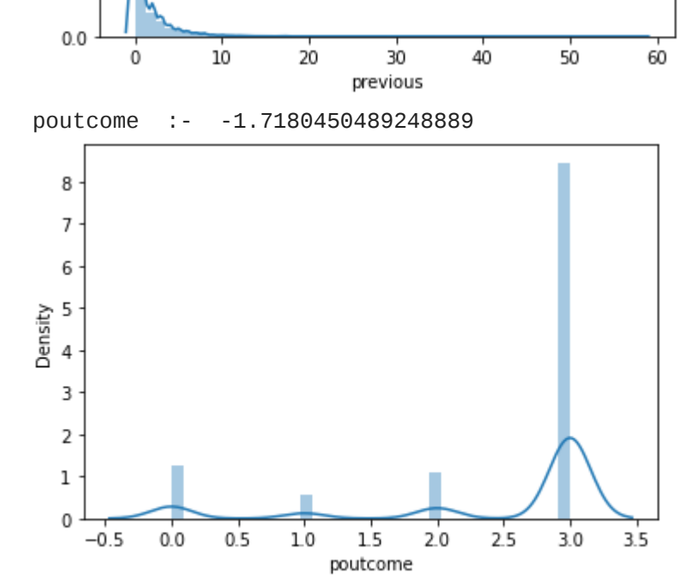
pdays :- 2.449656381764196



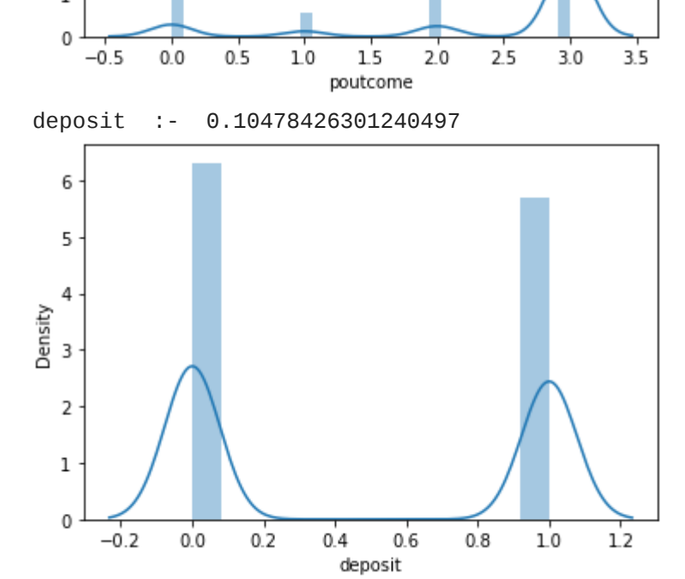
previous :- 7.334312390838451



poutcome :- -1.7180450489248889



deposit :- 0.10478426301240497



In [115]:
```python
df.skew()
```

Out[115]:
```
age          0.862780
job          0.184306
marital     -0.173968
education    0.115543
default      7.966983
balance      8.224619
housing      0.107678
loan         2.190201
contact      1.182309
day          0.111330
month       -0.322998
duration     2.143695
campaign     5.545578
pdays        2.449986
previous     7.335298
poutcome    -1.718276
deposit      0.104798
dtype: float64
```

In [116]:
```python
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
```

In [117]:
```python
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report
from sklearn.model_selection import train_test_split ,cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier , AdaBoostClassifier   ,GradientBoostingClassifier
from xgboost import XGBClassifier
```

In [118]:
```python
xtrain,xtest,ytrain,ytest=train_test_split(x,y,random_state=0,test_size=0.30)
```

In [119]:
```python
models=[('logistic Regression         :- ',LogisticRegression()),
        ('KNeighbors Classifier       :- ',KNeighborsClassifier()),
        ('Deceision Tree              :- ',DecisionTreeClassifier()),
        ('Support vectore Classifier  :- ',SVC()),
        ('Random Forest Clasifier     :- ',RandomForestClassifier()),
        ('Ada Boosting Classifier     :- ',AdaBoostClassifier()),
        ('Gradient Boosting Classifier :- ' ,GradientBoostingClassifier()),
        ('Xtreame Boosting Classifier  :- ',XGBClassifier())

       ]
```

In [120]:
```python
accuracy=[]
for name,model in models:
    print(name)
    model.fit(xtrain,ytrain)
    ypred=model.predict(xtest)
    print('Confusion matrix  :-  ')
    print(confusion_matrix(ytest,ypred))
    print(classification_report(ytest,ypred))
    accuracy.append(accuracy_score(ytest,ypred))
```

```
logistic Regression         :-
Confusion matrix  :-
[[1436  356]
 [ 453 1104]]
              precision    recall  f1-score   support

           0       0.76      0.80      0.78      1792
           1       0.76      0.71      0.73      1557

    accuracy                           0.76      3349
   macro avg       0.76      0.76      0.76      3349
weighted avg       0.76      0.76      0.76      3349

KNeighbors Classifier       :-
Confusion matrix  :-
[[1352  440]
 [ 446 1111]]
              precision    recall  f1-score   support

           0       0.75      0.75      0.75      1792
           1       0.72      0.71      0.71      1557

    accuracy                           0.74      3349
   macro avg       0.73      0.73      0.73      3349
weighted avg       0.74      0.74      0.74      3349

Deceision Tree              :-
Confusion matrix  :-
[[1404  388]
 [ 393 1164]]
              precision    recall  f1-score   support

           0       0.78      0.78      0.78      1792
           1       0.75      0.75      0.75      1557

    accuracy                           0.77      3349
   macro avg       0.77      0.77      0.77      3349
weighted avg       0.77      0.77      0.77      3349

Support vectore Classifier  :-
Confusion matrix  :-
[[1476  316]
 [ 557 1000]]
              precision    recall  f1-score   support

           0       0.73      0.82      0.77      1792
           1       0.76      0.64      0.70      1557

    accuracy                           0.74      3349
   macro avg       0.74      0.73      0.73      3349
weighted avg       0.74      0.74      0.74      3349

Random Forest Clasifier     :-
Confusion matrix  :-
[[1493  299]
 [ 219 1338]]
              precision    recall  f1-score   support

           0       0.87      0.83      0.85      1792
           1       0.82      0.86      0.84      1557

    accuracy                           0.85      3349
   macro avg       0.84      0.85      0.84      3349
weighted avg       0.85      0.85      0.85      3349

Ada Boosting Classifier     :-
Confusion matrix  :-
[[1510  282]
 [ 306 1251]]
              precision    recall  f1-score   support

           0       0.83      0.84      0.84      1792
           1       0.82      0.80      0.81      1557

    accuracy                           0.82      3349
   macro avg       0.82      0.82      0.82      3349
weighted avg       0.82      0.82      0.82      3349

Gradient Boosting Classifier :-
Confusion matrix  :-
[[1504  288]
 [ 227 1330]]
              precision    recall  f1-score   support

           0       0.87      0.84      0.85      1792
           1       0.82      0.85      0.84      1557

    accuracy                           0.85      3349
   macro avg       0.85      0.85      0.85      3349
weighted avg       0.85      0.85      0.85      3349

Xtreame Boosting Classifier  :-
[23:48:03] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
Confusion matrix  :-
[[1516  276]
 [ 225 1332]]
              precision    recall  f1-score   support

           0       0.87      0.85      0.86      1792
           1       0.83      0.86      0.84      1557

    accuracy                           0.85      3349
   macro avg       0.85      0.85      0.85      3349
weighted avg       0.85      0.85      0.85      3349
```

In [121]:
```python
print('Mean accuracy of all the models   :-  ',np.mean(accuracy)*100)
```

```
Mean accuracy of all the models   :-   79.57972529113168
```

In [ ]:

In [122]:
```python
xgb=XGBClassifier(booster='gbtree',gamma=12,reg_lambda =10.0,random_state=1)
xgb.fit(xtrain,ytrain)
ypred=xgb.predict(xtest)
```

```
[23:48:03] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
trainig score   :-   85.74171253039805
testing score   :-   84.77157360406092
```

In [126]:
```python
print('trainig score   :-   ',xgb.score(xtrain,ytrain)*100)
print('testing score   :-   ',xgb.score(xtest,ytest)*100)
```

```
trainig score   :-   85.74171253039805
testing score   :-   84.77157360406092
```

In [123]:
```python
c=cross_val_score(xgb,x,y,cv=10)
c
```

```
[23:48:04] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:04] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:05] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:05] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:05] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:06] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:06] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:07] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:08] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
[23:48:09] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

Out[123]:
```
array([0.75917637, 0.84059714, 0.75179211, 0.80824373, 0.79032258,
       0.83064516, 0.85663082, 0.85573477, 0.83422939, 0.83870968])
```

In [124]:
```python
print('Cross Validation Value :-   ',c.mean()*100)
```

```
Cross Validation Value :-   81.75081744175226
```

In [ ]: