



✓ 0s  `import numpy as np`

✓ 0s  `import pandas as pd`

✓ 0s  `import os
for dirname, _, filenames in os.walk(
 for filename in filenames:
 print(os.path.join(dirname,`

✓ 0s  `import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import
from sklearn.preprocessing import
from sklearn.neighbors import KNeig
from sklearn.preprocessing import l
from sklearn.metrics import accurac
from sklearn.metrics import mean_sc
from matplotlib.colors import Liste`

✓ 0s  `file_path = ('/content/bank-addition
df = pd.read_csv(file_path, sep=';')`

✓ 0s  `df`

✓
0s

df



age

job

marital

0

56

housemaid

married

1

57

services

married

2

37

services

married

3

40

admin.

married

4

56

services

married

...

...

...

...

41183

73

retired

married

pro

41184

46

blue-collar

married

pro

41185

56

retired

married

41186

44

technician

married

pro

41187

74

retired

married

pro

41188 rows × 21 columns

✓
0s

[69] df.isnull().sum()



0



[69] 0
0s



age 0

job 0

marital 0

education 0

default 0

housing 0

loan 0

contact 0

month 0

day_of_week 0

duration 0

campaign 0

pdays 0

previous 0

poutcome 0

emp.var.rate 0

cons.price.idx 0

cons.conf.idx 0



0s

`df.describe()`

age

duration

count

41188.00000

41188.000000

41

mean

40.02406

258.285010

std

10.42125

259.279249

min

17.00000

0.000000

25%

32.00000

102.000000

50%

38.00000

180.000000

75%

47.00000

319.000000

max

98.00000

4918.000000

0s

`df.y.value_counts()`

count

y

no

36548

yes

4640

dtype: int64

✓
0s

count



y

no 36548

yes 4640

dtype: int64

✓
0s

```
no=df[df['y']=='no']  
no.age.describe()
```



age

count 36548.000000

mean 39.911185

std 9.898132

min 17.000000

25% 32.000000

50% 38.000000

75% 47.000000

max 95.000000

dtype: float64

✓
0s

```
yes=df[df['y']=='yes']  
yes.age.describe()
```



age

count	4640.000000
-------	-------------

mean	40.913147
------	-----------

std	13.837476
-----	-----------

min	17.000000
-----	-----------

25%	31.000000
-----	-----------

50%	37.000000
-----	-----------

75%	50.000000
-----	-----------

max	98.000000
-----	-----------

dtype: float64

✓
0s

```
le=no.sample(n=4640,random_state=1)
```

✓
0s

```
[75] new_data = pd.concat([no_sample, ye  
new_data
```



age

job

marital

20507

35

self-
employed

married

0s

✓

0s

```
[75] new_data = pd.concat([no_sample, ye  
new_data
```



age

job

marital

20507

35

self-
employed

married

33693

35

self-
employed

single

641

29

admin.

married

10478

37

admin.

married

1582

44

technician

married

pro

...

...

...

...

41174

62

retired

married

41178

62

retired

married

41181

37

admin.

married

41183

73

retired

married

pro

41186

44

technician

married

pro

9280 rows × 21 columns

✓
0s

```
import seaborn as sns  
import matplotlib.pyplot as plt
```

✓
0s

```
[81] from sklearn.preprocessing import LabelEncoder  
     encoder = LabelEncoder()  
     data_df = new_data.apply(encoder.fit_transform, axis=1)  
     data_df
```



	age	job	marital	educati
--	-----	-----	---------	---------

20507	18	6	1
-------	----	---	---

33693	18	6	2
-------	----	---	---

641	12	0	1
-----	----	---	---

10478	20	0	1
-------	----	---	---

1582	27	9	1
------	----	---	---

...
-----	-----	-----	-----

41174	45	5	1
-------	----	---	---

41178	45	5	1
-------	----	---	---

41181	20	0	1
-------	----	---	---

41183	56	5	1
-------	----	---	---

41186	27	9	1
-------	----	---	---

9280 rows × 21 columns

✓
0s`data_df.y.value_counts()`**count****y****0** 4640**1** 4640**dtype:** int64✓
0s

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
normalized_data = scaler.fit_transform(data_df)
normalized_data
```



```
array([[0.24324324, 0.54545455,
        0.33333333, ..., 0.9829932 , 1.
        ,
        0.
        ],
       [0.24324324, 0.54545455,
        0.66666667, ..., 0.67006803, 0.6
        ,
        0.
        ],
       [0.16216216, 0.
        ,
        0.33333333, ..., 0.9047619 , 0.8
        ,
        0.
        ],
       ...,
       [0.27027027, 0.
        ,
        0.33333333, ..., 0.55102041, 0.]])
```

+ <> + T



RAM



Disk



0s



```
import numpy as np
corre = np.corrcoef(normalized_data)
```

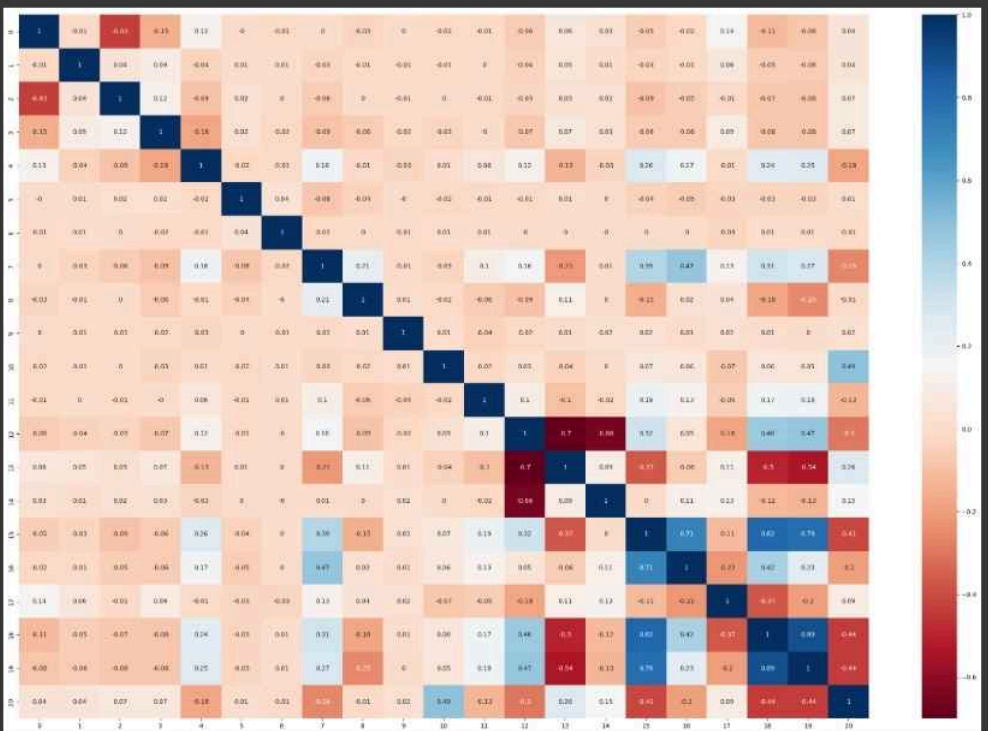
3s



```
plt.figure(figsize=(30, 20))
corre_rounded = np.round(corre, 2)
sns.heatmap(corre_rounded, cmap='Rd
```



<Axes: >



0s

```
[88] w=data_df.iloc[:,[4,7,10,11,12,15,1
x=scaler.fit_transform(w)
x
```



```
array([[0.          , 0.          ,
0.07692308, ..., 0.52
0.9829932 ,
1.          ],
[0.          , 0.          ,
0.07478632, ..., 0.32
0.67006803,
```

✓
0s

```
[88] w=data_df.iloc[:,[4,7,10,11,12,15,16]]
      x=scaler.fit_transform(w)
      x
```



```
array([[0.          , 0.
0.07692308, ..., 0.52
0.9829932 ,
1.          ],
[0.          , 0.
0.07478632, ..., 0.32
0.67006803,
0.6          ],
[0.          , 1.
0.01780627, ..., 0.72
0.9047619 ,
0.8          ],
...,
[0.          , 0.
0.19729345, ..., 1.
0.55102041,
0.          ],
[0.          , 0.
0.23504274, ..., 1.
0.55102041,
0.          ],
[0.          , 0.
0.31196581, ..., 1.
0.55102041,
0.          ]])
```

0s



```
y=data_df.iloc[:,-1]
```

✓
0s

```
y=data_df.iloc[:,-1]  
y
```



y

20507 0

33693 0

641 0

10478 0

1582 0

... ...

41174 1

41178 1

41181 1

41183 1

41186 1

9280 rows × 1 columns

dtype: int64

✓
0s

```
from sklearn.model_selection import  
from sklearn.svm import SVC
```

✓
0s

```
from sklearn.model_selection import  
from sklearn.svm import SVC  
from sklearn.linear_model import Lc  
from sklearn.neighbors import KNeig  
  
from sklearn.tree import DecisionTr  
from sklearn.ensemble import Randon  
from sklearn.naive_bayes import Gau  
from sklearn.metrics import accurac
```

✓
0s

```
(x,y,test_size=0.2,random_state=22)
```

✓
0s

```
tree=DecisionTreeClassifier()  
tree.fit(x_train, y_train)  
pred_3=tree.predict(x_test)
```

✓
0s

```
acc=accuracy_score(y_test,pred_3)  
print('accuracy is : ',acc)
```



```
accuracy is : 0.8238146551724138
```