\

DATA WAREHOUSING AND DATA MINING

CSE4005

LAB5&6

ASSIGNMENT

Name-Gouthami M

Registration No.-19BCD7148

Slot- L27+L28+L9+L10+L41+L42

Date-29/10/2021

Faculty- Mr. Koteswararao Chitipireddi

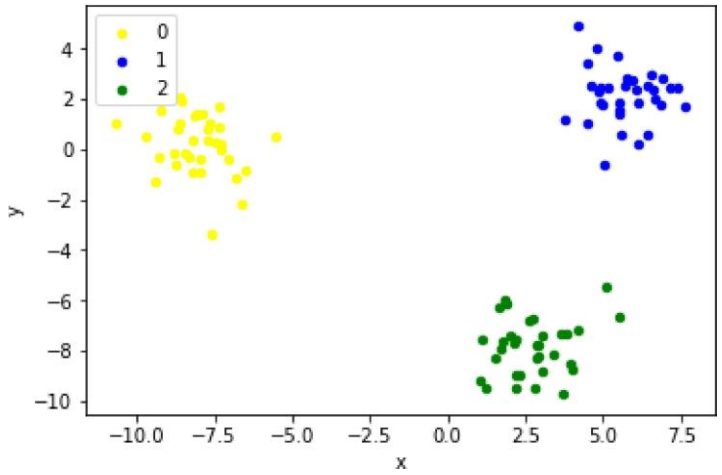
\

#EXP-5

#1

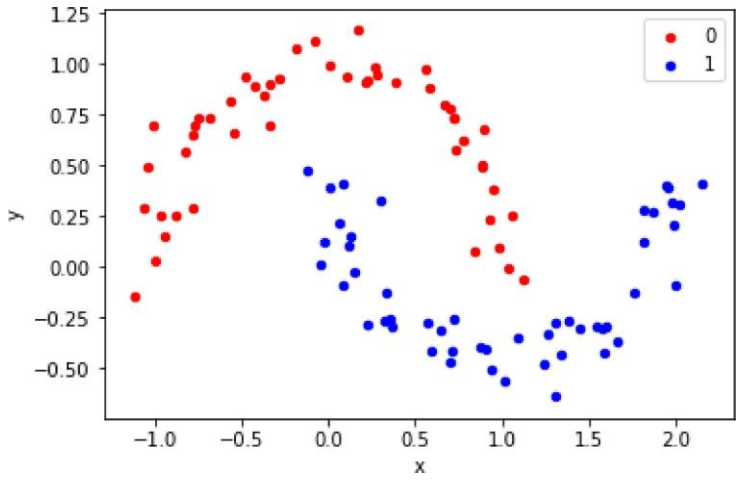
In [40]:

```
from sklearn.datasets import make_blobs
from matplotlib import pyplot
from pandas import DataFrame
# generate 2d c L assifi cat i on dataset
X, y = make_blobs(n_samples=100, centers=3, n_features=2)
#  s catter p Lot dots co L or ed by c Lass va Lue
df = DataFrame(dict(x=X[:,0], y=X[:,1], 1abel=y))
colors = {0:'yellow', 1:'blue', 2:'green'}
fig, ax = pyplot.subplots()
grouped =df.groupby('label')
for key, group in grouped:
    group.plot(ax=ax, kind='scatter', x='x', y='y', label=key, color=colors[key])
pyplot.show()
# generate 2d c Pass i fi cat i on dataset
X, y = make_blobs(n_samples=100, centers=3, n_features=2)
```



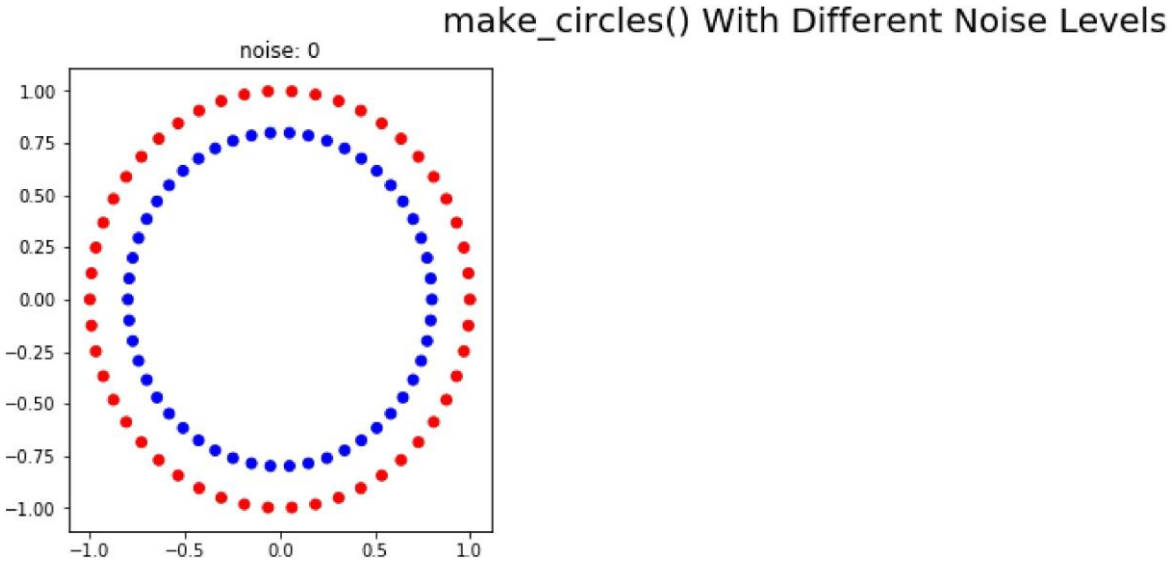
In [33]:

```
from sklearn.datasets import make_moons
from matplotlib import pyplot
from pandas import DataFrame
it genera te 2d c Lass i i cat:i on dat:ase t
X, y = make_moons(n_samples=100, noise=0.1)
# s caller pL of, dots coL ored by c Lass vaLue
df = DataFrame(dict(x=X[:,0], y=X[:,1], label=y))
colors = {0:'red', 1:'blue'}
fig, ax = pyplot.subplots()
grouped =df.groupby('label')
for key, group in grouped:
    group.plot(ax=ax, kind='scatter', x='x', y='y', label=key, color=colors[key])
pyplot.show()
```



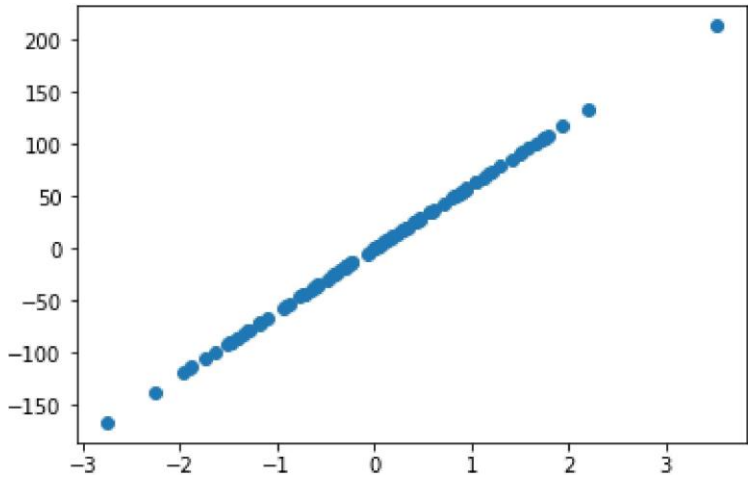
In [43]:

```
fig,ax   plt.subplots(nrows=1, ncols=1,figsize=(16,5))
plt_ind_list = np.arange(3)+131

for noise,plt_ind in zip([0],plt_ind_list):
    x, label   dt.make_circles(noise=noise,random_state=rand_state)

    plt.subplot(plt_ind)
    my_scatter_plot = plt.scatter(x[:,0],
                                  x[:,1],
                                  c=label,
                                  vmin=min(label),
                                  vmax=max(label),
                                  cmap=color_map_discrete)
    plt.title('noise: '+str(noise))

fig.subplots_adjust(hspace=0.3,wspace=.3)
plt.suptitle('make_circles() With Different Noise Levels',fontsize=20)
plt.show()
```



In [37]

```
from sklearn.datasets import make_regression
from matplotlib import pyplot
# generate regres s i on dataset
X, y = make_regression(n_samples=100, n_features=1, noise=0.1)
it p Lot   regress i on datas et:
pyplot.scatter(X,y)
pyplot.show()

X, y = make_regression(n_samples=100, n_features=1, noise=0.1)
```



question 2

In [13]

```
pip install Faker
```

```
Collecting Faker
  Downloading https://files.pythonhosted.org/packages/19/1c/0d3248616f697230
305bf910669e3cf11898962a6a91757df6c07c1488f4/Faker-9.7.1-py3-none-any.whl (h
ttps://files.pythonhosted.org/packages/19/1c/0d32486l6f697230305bf9l0669e3cf
11898962a6a91757df6c07c1488f4/Faker-9.7.1-py3-none-any.whl) (1.2MB)
Collecting text-unidecode==1.3 (from Faker)
  Downloading https://files.pythonhosted.org/packages/a6/a5/c0b6468d3824fe3f
de30dbb5e1f687b291608f9473681bbf7dabbf5a87d7/text_unidecode-1.3-py2.py3-none
-any.whl (https://files.pythonhosted.org/packages/a6/a5/c0b6468d3824fe3fde30
dbb5e1f687b291608f9473681bbf7dabbf5a87d7/text_unidecode-1.3-py2.py3-none-an
y.whl) (78kB)
Collecting typing-extensions>=3.10.0.2; python_version < "3.8" (from Faker)
  Downloading https://files.pythonhosted.org/packages/74/60/18783336cc7fcdd9
5dae91d73477830aa53f5d3181ae4fe20491d7fc3199/typing_extensions-3.10.0.2-py3-
none-any.whl (https://files.pythonhosted.org/packages/74/60/18783336cc7fcdd9
5dae91d73477830aa53f5d3181ae4fe20491d7fc3199/typing_extensions-3.10.0.2-py3-
none-any.whl)
Requirement already satisfied: python-dateutil>=2.4 in c:\programdata\anacon
da3\lib\site-packages (from Faker) (2.8.0)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site
-packages (from python-dateutil>=2.4->Faker) (1.12.0)
Installing collected packages: text-unidecode, typing-extensions, Faker
Successfully installed Faker-9.7.1 text-unidecode-1.3 typing-extensions-3.1
0.0.2
Note: you may need to restart the kernel to use updated packages.
```

In [19]:

```
import csv
from faker import Faker
import datetime

def datagenerate(records, headers):
    fake = Faker('en_US')
    fakel = Faker('en_GB')  # To generore phone numders
    with open("People_data.csv", 'wt') as csvFile:
        writer  csv.DictWriter(csvFile, fieldnames=headers)
        writer.writeheader()
        for i in range(records):
            full_name = fake.name()
            FLname = full_name.split(" ")
            Fname = FLname[0]
            Lname = FLname[1]
            domain_name = "@testDomain.com"
            userId = Fname +"."+ Lname + domain_name

            writer.writerow({
                    "Customer id" : userId,

                    "Name": fake.name(),
                    "Birth Date" fake.date(pattern="%d-%m-%Y", end_datetime=datetime.date
                    "Phone Number" : fakel.phone_number(),
                    "Email Id": fake.email(),
                    "Address" : fake.address(),
                    "Zip Code" : fake.zipcode(),
                    "City" : fake.city(),
                    "State" : fake.state(),
                    "Country" : fake.country(),
                    "Year":fake.year(),



if   name   ==    ma1n
    records = 10000
    headers =["Customer id" , "Name", "Birth Date", "Phone Number", "Email Id",
              "Address", "Zip Code", "City","State", "Country", "Year"]
    datagenerate(records, headers)
    print("CSV generation complete!")
```

```
CSV generation complete!
```

In [ ]:

In [1]:

```
#1
```

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as  sns
import matplotlib.pyplot as plt
```

In [4]:

```
df.head()
```

Out[4]:

	age	job	marital	education	default	balance	housing	loan	contact	day	mo	
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	m	
1	44	technician	single	secondary	no	29	yes	no	unknown	5	m	
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	m	
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	m	
4	33	unknown	single	unknown	no	1	no	no	unknown	5	m	

In [5]:

```
df['marital'].value_counts()
```

Out[5]:

```
married     27214
single      12790
divorced     5207
Name: marital, dtype: int64
```

In [6]:

```
map1 = {'single':0,'married':1,'divorced':1}
```

In [7]:

```
df['marital'] = df['marital'].map(map1)
```

In [8]:

```
df.head(3)
```

Out[8]:

	age	job	marital	education	default	balance	housing	loan	contact	day	mo
0	58	management	1	tertiary	no	2143	yes	no	unknown	5	m
1	44	technician	0	secondary	no	29	yes	no	unknown	5	m
2	33	entrepreneur	1	secondary	no	2	yes	yes	unknown	5	m

In [9]:

```
map2 = {'no':0,'yes':1}
```

In [10]:

```
df['housing'] = df['housing'].map(map2)
```

In [11]:

```
df.head(3)
```

Out[11]:

	age	job	marital	education	default	balance	housing	loan	contact	day	mo
0	58	management	1	tertiary	no	2143	1	no	unknown	5	m
1	44	technician	0	secondary	no	29	1	no	unknown	5	m
2	33	entrepreneur	1	secondary	no	2	1	yes	unknown	5	m

In [12]:

```
map3 = {'no':0,'yes':1}
```

In [13]:

```
df['loan'] = df['loan'].map(map3)
```

In [14]:

```
df.head(3)
```

Out[14]:

	age	job	marital	education	default	balance	housing	loan	contact	day	mo
0	58	management	1	tertiary	no	2143	1	0	unknown	5	m
1	44	technician	0	secondary	no	29	1	0	unknown	5	m
2	33	entrepreneur	1	secondary	no	2	1	1	unknown	5	m

In [15]:

```
map4 = {'blue-collar':3,
'management':0,
'technician':1,
'admin.':5,
'services':6,
'retired':4,
'self-employed':7,
'entrepreneur':2,
'unemployed':8,
'housemaid':9,
'student':10,
'unknown':np.nan}
```

In [16]:

```
df['job'] = df['job'].map(map4)
```

In [17]:

```
df.head(3)
```

Out[17]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	dura
0	58	0.0	1	tertiary	no	2143	1	0	unknown	5	may	
1	44	1.0	0	secondary	no	29	1	0	unknown	5	may	
2	33	2.0	1	secondary	no	2	1	1	unknown	5	may	

In [18]:

```
map5 = {'secondary':1,
'tertiary':0,
'primary':2,
'unknown':np.nan}
```

In [19]:

```
df['education'] = df['education'].map(map5)
```

In [20]:

```
df.head(3)
```

Out[20]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	dura
0	58	0.0	1	0.0	no	2143	1	0	unknown	5	may	
1	44	1.0	0	1.0	no	29	1	0	unknown	5	may	
2	33	2.0	1	1.0	no	2	1	1	unknown	5	may	

In [21]:

```
map6 = {'no':0,'yes':1}
```

In [22]:

```
df['default'] = df['default'].map(map6)
```

In [23]:

```
df.head(3)
```

Out[23]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	dura
0	58	0.0	1	0.0	0	2143	1	0	unknown	5	may	
1	44	1.0	0	1.0	0	29	1	0	unknown	5	may	
2	33	2.0	1	1.0	0	2	1	1	unknown	5	may	

In [24]:

```
df['contact'] = df['contact'].map({'unknown':np.nan,'telephone':0,'cellular':1})
```

In [25]:

```
df['poutcome'] = df['poutcome'].map({'unknown':np.nan,'failure':0,'other':1,'success':2
})
```

In [26]:

```
df.head(3)
```

Out[26]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	durat
0	58	0.0	1	0.0	0	2143	1	0	NaN	5	may	
1	44	1.0	0	1.0	0	29	1	0	NaN	5	may	
2	33	2.0	1	1.0	0	2	1	1	NaN	5	may	

In [27]:

```
df['month'] = df['month'].map({
    'may'  :  5,
'jul'       :7,
'aug':      8,
'jun'  :    6,
 'nov'   :  11,
'apr'       :4,
'feb'     : 2,
'jan'       : 1,
'oct'        :10,
'sep'        :9,
'mar'       : 3,
'dec'        :12
})
```

In [28]:

```
df.head(3)
```

Out[28]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	durat
0	58	0.0	1	0.0	0	2143	1	0	NaN	5	5	
1	44	1.0	0	1.0	0	29	1	0	NaN	5	5	
2	33	2.0	1	1.0	0	2	1	1	NaN	5	5	

In [29]:

```
df['Target'] = df['Target'].map({'no':0,'yes':1})
```

In [30]:

```
df.head(3)
```

Out[30]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	durat
0	58	0.0	1	0.0	0	2143	1	0	NaN	5	5	
1	44	1.0	0	1.0	0	29	1	0	NaN	5	5	
2	33	2.0	1	1.0	0	2	1	1	NaN	5	5	

In [31]:

```
import csv

df.to_csv('preprocessed.csv',index=True)
```

In [32]:

```
#2
plt.figure(figsize=(12,6))

sns.heatmap(data=df.corr(),cmap='coolwarm')
```

Out[32]:

<matplotlib.axes._subplots.AxesSubplot at 0x1aa196b32b0>