NP DM CS 2024

31 Jan 2024 Practice

In [ ]:
```
from sklearn import datasets
```

In [ ]:
```
datasets_all=datasets.__all__
```

In [ ]:
```
datasets_all
```

In [ ]:
```
from sklearn.datasets import load_wine
```

In [ ]:
```
wine_data=load_wine()
```

In [ ]:
```
wine_data
```

In [ ]:
```
import pandas as pd
```

In [ ]:
```
wine_df=pd.DataFrame(data=wine_data.data,columns=wine_data.feature_names)
```

In [ ]:
```
wine_df['class']=wine_data.target
```

In [ ]:
```
wine_df.head()
```

In [ ]:
```
wine_df.tail()
```

In [ ]:
```
wine_df.info()
```

In [ ]:
```
wine_df.shape
```

In [ ]:
```
wine_df.describe()
```

In [ ]:
```
wine_df.isnull().any()
```

In [ ]:
```
wine_df.duplicated()
```

In [ ]:
```
##example for preprocessing
```

In [ ]:
```
import numpy as np
import pandas as pd
```

In [ ]:
```
data = {
    'Name': ['John', 'Anna', 'Peter', 'Lily',np.nan],
    'Age': [25, 30, 335, 4, np.nan],
    'City': ['New York', 'Paris', np.nan, 'Tokyo', np.nan]
}
```

In [ ]:
```
data=pd.DataFrame(data)
```

In [ ]:
```
data
```

In [ ]:
```
data.head()
```

In [ ]:
```
data.info()
```

In [ ]:
```
data.shape
```

In [ ]:
```
data.isna().any()
```

In [ ]:
```
data.isna().sum()
```

In [ ]:
```
data.duplicated()
```

In [ ]:
```
data.describe()
```

In [ ]:
```
data
```

In [ ]:
```
data1=data.dropna(thresh=2)
```

In [ ]:
```
data=data.iloc[:,:].replace(30,np.inf) #adding inf in data to clean later
```

In [ ]:
```
data
```

In [ ]:
```
data=data.iloc[:,:].replace(np.inf,np.nan)
print(data)
```

In [ ]:
```
rule1=data['Age'].apply(lambda x:True if x>18 and x<100 else False)
rule2=data['City'].apply(lambda x:True if x=='Paris' or x=='Tokyo' else False)
```

In [ ]:
```
rule1
```

In [ ]:
```
rule2
```

In [ ]:
```
Rules=pd.DataFrame({"Rule 1":rule1,"Rule 2" : rule2})
```

In [ ]:
```
Rules
```

In [ ]:
```
Rules.astype(int)
```

In [ ]:
```
Rules=Rules.astype(int)
print(Rules)
```

In [ ]:
```
print("number of rules violated : ",len(Rules)-Rules["Rule 1"].sum())
print("count of both \n", Rules["Rule 1"].value_counts())
```

In [ ]:
```
print("number of rules violated :", len(Rules)-Rules["Rule 2"].sum())
```

In [ ]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
plt.figure()
Rules.apply(lambda x:len(x)-x.sum()).plot(kind='bar')
plt.xlabel="Rules"
plt.ylabel="Number of records that violates the Rules"
```

In [ ]:
```

```