1.2 Multivariate filter Methods:

Multivariate filter methods are capable of removing redundant features from the data since they take the mutual relationship between the features into account.

Multivariate filter methods can be used to remove duplicate and correlated features from the data.

Removing duplicate Features(Redundant Features)

Duplicate features are the features that have similar values. Duplicate features do not add any value to algorithm training, rather they add overhead and unnecessary delay to the training time. Therefore, it is always recommended to remove the duplicate features from the dataset before training

In [2]:

```
#  importing required libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
```

In [3]:

```
# read the data file
dup_data = pd.read_csv('train.csv',nrows = 20000)
```

In [4]:

```
dup_data.shape
```

Out[4]:

```
(20000, 371)
```

In [5]:

```
# split the data into training and testing sets
train_features , test_features ,train_labels, test_labels = train_test_split(
 dup_data.drop(labels = ['TARGET'] , axis =1),
 dup_data['TARGET'],
 test_size = 0.2,
 random_state = 41)
```

In [6]:

```
print(train_features.shape)
train_features_t = train_features.T
train_features_t.shape
```

```
(16000, 370)
```

Out[6]:

```
(370, 16000)
```

In the script above we take the transpose of our training data and store it in the train_features_T dataframe. Our initial training set contains 16000 rows and 370 columns, if you take a look at the shape of the transposed training set, you will see that it contains 370 rows and 16000 columns.

Luckily, in pandas we have duplicated() method which can help us find duplicate rows from the dataframe.

In [11]:

```
#  to know the total no:of duplicated features we use sum() method
print(train_features_t.duplicated().sum())
```

```
94
```

we can drop the duplicate rows using the drop_duplicates() method. If you pass the string value first to the keep parameter of the drop_duplicates() method, all the duplicate rows will be dropped except the first copy

In [12]:

```
unique_features = train_features_t.drop_duplicates(keep = 'first')
unique_features.shape
```

Out[12]:

```
(276, 16000)
```

In [13]:

```
unique_features = unique_features.T
```

In [14]:

```
unique_features.shape
```

Out[14]:

```
(16000, 276)
```

In [15]:

```
dup_features= [col for col in train_features.columns
               if col not in unique_features.columns]
len(dup_features)
```

Out[15]:

94

REMOVING CORRELATED FEATURES

Correlation between the output observations and the input features is very important and such features should be retained. However, if two or more than two features are mutually correlated, they convey redundant information to the model and hence only one of the correlated features should be retained to reduce the number of features.

In [7]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```

In [8]:

```
paribas_data = pd.read_csv('traincorr.csv', nrows = 20000)
paribas_data.shape
```

Out[8]:

(20000, 133)

To find the correlation, we only need the numerical features in our dataset. In order to filter out all the features, except the numeric ones, we need to preprocess our data.

In [9]:

```
# DATA PERPROCESSING
num_colums = ['int16','int32','int64','float16','float32','float64']

numerical_columns = list(paribas_data.select_dtypes(include = num_colums).columns)
paribas_data = paribas_data[numerical_columns]
```

In [10]:

```
paribas_data.shape
```

Out[10]:

(20000, 114)

In [11]:

```
train_features , test_features , train_labels , test_labels =  train_test_split(
paribas_data.drop(labels = ['target','ID'] , axis = 1),
paribas_data['target'],
test_size = 0.2,
random_state= 41
)
```

Removing Correlated Features using corr() Method:

To remove the correlated features, we can make use of the corr() method of the pandas dataframe. The corr() method returns a correlation matrix containing correlation between all the columns of the dataframe

Pearson's Correlation: It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. Pearson's correlation is given as:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

In [13]:

```
correlated_features = set()
correlation_matrix = paribas_data.corr()
```

In [14]:

```
for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i,j])>0.8:
            colname = correlation_matrix.columns[i]
            correlated_features.add(colname)
```

In [15]:

```
len(correlated_features)
```

Out[15]:

```
55
```

In [16]:

```
train_features.drop(labels= correlated_features , axis =1 ,inplace = True)
test_features.drop(labels = correlated_features , axis = 1, inplace = True)
```

In [17]:

```
train_features.shape, test_features.shape
```

Out[17]:

```
((16000, 57), (4000, 57))
```

LINKS FOR THE BUILT-IN METHODS DOCCUMENTATIONS USED IN ABOVE PROGRAM

read_csv() : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html)

train_test_split() : https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)

VarianceThreshold() : https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html (https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html)

transform() : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.transform.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.transform.html)

fit() : https://scikit-learn.org/stable/modules/generated/sklearn.svm.libsvm.fit.html (https://scikit-learn.org/stable/modules/generated/sklearn.svm.libsvm.fit.html)

drop() : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html)

corr() : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html)

duliacated() :https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html)

drop_duplicates(): https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_duplicates.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_duplicates.html)