Question 1

In [1]:
```
import numpy as np
from sklearn.model_selection import cross_val_score
bikeDataSet = np.genfromtxt('./Bike-Sharing-Dataset/hour.csv', delimiter='
X = bikeDataSet[1:,2:-1]
y = bikeDataSet[1:,-1]
from sklearn import linear_model
estimator = linear_model.LinearRegression()
score = cross_val_score(estimator, X, y).mean()
print(f"Score = {score:.3f}.")
```

```
Score = 1.000.
```

In [2]:
```
import numpy as np
from sklearn.model_selection import cross_val_score
bikeDataSet = np.genfromtxt('./Bike-Sharing-Dataset/hour.csv', delimiter='
X=np.random.rand(X.shape[0],4)
y = bikeDataSet[1:,-1]
from sklearn import linear_model
estimator = linear_model.LinearRegression()
score = cross_val_score(estimator, X, y).mean()
print(f"Score = {score:.3f}.")
```

```
Score = -0.280.
```

In [3]:
```
import numpy as np
from sklearn.model_selection import cross_val_score
bikeDataSet = np.genfromtxt('./Bike-Sharing-Dataset/hour.csv', delimiter='
X=np.random.rand(X.shape[0],4)
y = bikeDataSet[1:,-1]
from sklearn import linear_model
estimator = linear_model.Lasso(alpha=0.1)
score = cross_val_score(estimator, X, y).mean()
print(f"Score = {score:.3f}.")
```

```
Score = -0.280.
```

In [4]:
```
import numpy as np
from sklearn.model_selection import cross_val_score
bikeDataSet = np.genfromtxt('./Bike-Sharing-Dataset/hour.csv', delimiter='
X = bikeDataSet[1:,2:-1]
y = bikeDataSet[1:,-1]
from sklearn import linear_model
estimator = linear_model.Lasso(alpha=0.1)
score = cross_val_score(estimator, X, y).mean()
print(f"Score = {score:.3f}.")
```

```
Score = 1.000.
```

Question 2

In [3]:
```
import pandas as pd
import time
from sklearn.preprocessing import LabelBinarizer
bikeDataFrame=pd.read_csv( './kddcup.data_10_percent.gz', header=None )
start_time=time.time()
start_time1=time.time()
# print(bikeDataFrame.head())
# print( bikeDataFrame.dtypes )
# bikeDataFrame[ 1 ] = bikeDataFrame[ 1 ].astype('category').cat.codes
# print(bikeDataFrame.head())
bikeDataFrame = pd.get_dummies(bikeDataFrame)
print(bikeDataFrame.head())
print(len(bikeDataFrame))
print("One hot encoding time is %s seconds" % (time.time() - start_time))
start_time = time.time()
bikeDataFrame_norm = (bikeDataFrame - bikeDataFrame.mean())/bikeDataFrame.
print(bikeDataFrame_norm)
print("Normalisation Time is %s seconds" % (time.time() - start_time))
print("Time for both together is %s seconds" %(time.time()-start_time1))
```

```
494019    -0.004499  -0.056805  -1.147533 -0.002012      -0.04456
494020    -0.004499  -0.056805  -1.147533 -0.002012      -0.04456

        41_warezclient.  41_warezmaster.
0             -0.045486        -0.006363
1             -0.045486        -0.006363
2             -0.045486        -0.006363
3             -0.045486        -0.006363
4             -0.045486        -0.006363
...                 ...              ...
494016        -0.045486        -0.006363
494017        -0.045486        -0.006363
494018        -0.045486        -0.006363
494019        -0.045486        -0.006363
494020        -0.045486        -0.006363

[494021 rows x 141 columns]
Normalisation Time is 1.0734434127807617 seconds
Time for both together is 1.6555554866790771 seconds
```

Question 3

In [6]:
```
from sklearn import datasets
from scipy.stats import describe
from sklearn.svm import SVC
alpha = 1
# Load the dataset to X and y
iris = datasets.load_iris()
X = iris.data
y = iris.target
X = X * alpha
estimator = SVC(kernel = 'linear')
score = cross_val_score(estimator, X, y).mean()
print(f"Score = {score:.3f}.")
```

```
Score = 0.980.
```

In [7]:
```
# highest value obtained when alpha is 1
```