Machine Learning Fundamentals- Lab-3

Name: Gaurav Prasanna
Reg No: 19BEC1315

Aim:

a) To show Data cleaning method with the help Numpy library and visualizing using Matplotlib with oil-spill dataset.
b) To show Data Duplication using Pandas library for Iris dataset.

Software Required:

1) Anaconda Navigator
2) Jupyter Notebook

Libraries Required: Numpy, Pandas, Matplotlib

Code and Outputs:

a) Data Cleaning using oil-spill dataset

In [2]:
```
import numpy
from numpy import loadtxt, unique
import matplotlib.pyplot as plt
```

In [3]:
```
data = loadtxt('oil-spill.csv', delimiter= ',')
data
```

Out[3]:
```
array([[1.00000e+00, 2.55800e+03, 1.50609e+03, ..., 6.57400e+01,
        7.95000e+00, 1.00000e+00],
       [2.00000e+00, 2.23250e+04, 7.91100e+01, ..., 6.57300e+01,
        6.26000e+00, 0.00000e+00],
       [3.00000e+00, 1.15000e+02, 1.44985e+03, ..., 6.58100e+01,
        7.84000e+00, 1.00000e+00],
       ...,
       [2.02000e+02, 1.40000e+01, 2.51400e+01, ..., 6.59100e+01,
        6.12000e+00, 0.00000e+00],
       [2.03000e+02, 1.00000e+01, 9.60000e+01, ..., 6.59700e+01,
        6.32000e+00, 0.00000e+00],
       [2.04000e+02, 1.10000e+01, 7.73000e+00, ..., 6.56500e+01,
        6.26000e+00, 0.00000e+00]])
```

In [4]:
```
range(data.shape[1])
```

Out[4]:
```
range(0, 50)
```

In [5]:
```
unique(data[:,0])
```

Out[5]:
```
array([  1.,   2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,  10.,  11.,
        12.,  13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,  21.,  22.,
        23.,  24.,  25.,  26.,  27.,  28.,  29.,  30.,  31.,  32.,  33.,
        34.,  35.,  36.,  37.,  38.,  39.,  40.,  41.,  42.,  43.,  44.,
        45.,  46.,  47.,  48.,  49.,  50.,  51.,  52.,  53.,  54.,  55.,
        56.,  57.,  58.,  59.,  60.,  61.,  62.,  63.,  64.,  65.,  66.,
        67.,  68.,  69.,  70.,  71.,  72.,  73.,  74.,  75.,  76.,  77.,
        78.,  79.,  80.,  81.,  82.,  83.,  84.,  85.,  86.,  87.,  88.,
        89.,  90.,  91.,  92.,  93.,  94.,  95.,  96.,  97.,  98.,  99.,
       100., 101., 102., 103., 104., 105., 106., 107., 108., 109., 110.,
       111., 112., 113., 114., 115., 116., 117., 118., 119., 120., 121.,
       122., 123., 124., 125., 126., 127., 128., 129., 130., 131., 132.,
       133., 134., 135., 136., 137., 138., 139., 140., 141., 142., 143.,
       144., 145., 146., 147., 148., 149., 150., 151., 152., 153., 154.,
       155., 156., 157., 158., 159., 160., 161., 162., 163., 164., 165.,
       166., 167., 168., 169., 170., 171., 172., 173., 174., 175., 176.,
       177., 178., 179., 180., 181., 182., 183., 184., 185., 186., 187.,
       188., 189., 190., 191., 192., 193., 194., 195., 196., 197., 198.,
       199., 200., 201., 202., 203., 204., 206., 207., 208., 215., 216.,
       225., 227., 228., 231., 232., 233., 235., 237., 244., 245., 247.,
       254., 257., 261., 266., 267., 269., 271., 272., 280., 281., 284.,
       310., 317., 321., 328., 332., 339., 352.])
```

In [6]:
```
for i in range(data.shape[1]):
    print(i, len(unique(data[:, i])))
```

```
0 238
1 297
2 927
```

In [7]:
```
import pandas as pd
```

In [19]:
```
df = pd.read_csv('oil-spill.csv', header=None)
print(df.nunique())
```

```
0     238
1     297
2     927
3     933
4     179
5     375
6     820
7     618
8     561
9      57
10    577
11     59
12     73
13    107
14     53
15     91
16    893
17    810
18    170
```

In [18]:
```
to_del = [i for i, v in enumerate(counts) if v == 1]
print(to_del)
df.drop(to_del, axis=1, inplace=True)
print(df.shape)
```

```
[22]
(937, 49)
```

In [12]:
```
data.shape[0]
```

Out[12]: 937

In [13]:
```
len(unique(data[:, 3]))
```

Out[13]: 933

In [14]:
```
for i in range(data.shape[1]):
    num = len(unique(data[:, i]))
    percentage = float(num)/ data.shape[0] * 100
    print('%d, %d, %.1f%%' % (i, num, percentage))
```

```
0, 238, 25.4%
1, 297, 31.7%
2, 927, 98.9%
3, 933, 99.6%
4, 179, 19.1%
5, 375, 40.0%
6, 820, 87.5%
7, 618, 66.0%
8, 561, 59.9%
```

In [20]:
```
to_del = [i for i, v in enumerate(counts) if (float(v)/ df.shape[0] * 100) < 1]
print(to_del)
df.drop(to_del, axis=1, inplace=True)
print(df.shape)
```

```
[21, 22, 24, 25, 26, 32, 36, 38, 39, 45, 49]
(937, 39)
```

In [21]:
```
from sklearn.feature_selection import VarianceThreshold

df = pd.read_csv('oil-spill.csv', header=None)
data = df.values
X = data[:, :-1]
y = data[:, -1]
print(X.shape, y.shape)
transform = VarianceThreshold()
X_sel = transform.fit_transform(X)
print(X_sel.shape)
```

```
(937, 49) (937,)
(937, 48)
```

In [22]:
```
from numpy import arange
thresholds = arange(0.0, 0.55, 0.05)
```

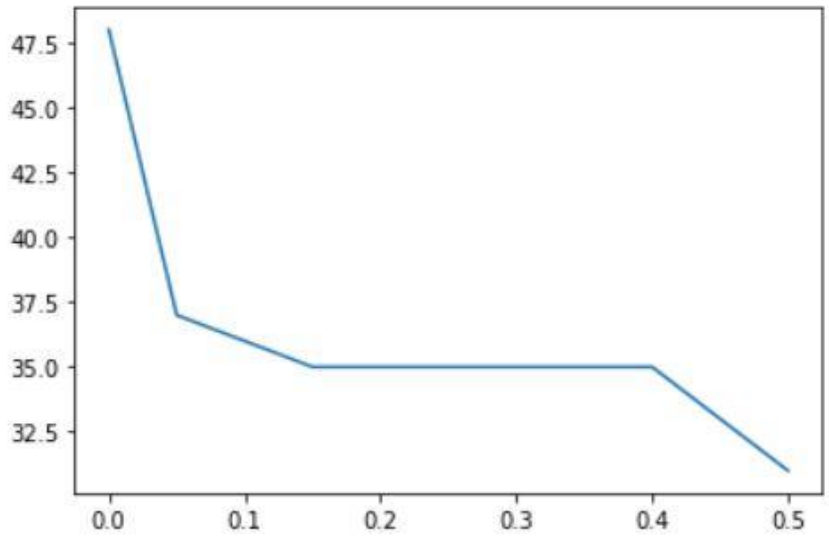
In [23]:
```
results = list()
for t in thresholds:
    transform = VarianceThreshold(threshold=t)
    X_sel = transform.fit_transform(X)
    n_features = X_sel.shape[1]
    print('>Threshold=%.2f, Features=%d' %(t, n_features))
    results.append(n_features)
```

```
>Threshold=0.00, Features=48
>Threshold=0.05, Features=37
>Threshold=0.10, Features=36
>Threshold=0.15, Features=35
>Threshold=0.20, Features=35
>Threshold=0.25, Features=35
>Threshold=0.30, Features=35
>Threshold=0.35, Features=35
>Threshold=0.40, Features=35
>Threshold=0.45, Features=33
>Threshold=0.50, Features=31
```

In [24]:
```
%matplotlib inline
```

In [25]:
```
plt.plot(thresholds, results)
```

Out[25]:
```
[<matplotlib.lines.Line2D at 0x1fcfb379460>]
```



b) Data Duplication in Iris Dataset

In [1]:
```
import pandas as pd

df = pd.read_csv('Iris.csv', header=None)
dups = df.duplicated()
print(dups.any())
```

```
False
```

In [2]:
```
print(df[dups])
print(df.shape)
```

```
Empty DataFrame
Columns: [0, 1, 2, 3, 4, 5]
Index: []
(151, 6)
```

In [3]:
```
df.drop_duplicates(inplace=True)
print(df.shape)
```

```
(151, 6)
```

Inference: So from the above two parts we understand the data cleaning and duplication using Numpy and Pandas, so various dataset to do necessary operations needed for data before building a machine learning model.

Result: Data cleaning and duplication is shown and visualized using different libraries in Jupyter Notebook.