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


dataset = pd.read_csv('/content/drive/My Drive/kddcup99_csv.csv')


dataset.head(5)
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment
0	0	tcp	http	SF	181	5450	0	0
1	0	tcp	http	SF	239	486	0	0
2	0	tcp	http	SF	235	1337	0	0
3	0	tcp	http	SF	219	1337	0	0
4	0	tcp	http	SF	217	2032	0	0

feature selection by using variance method

```
print(dataset.var()['src_bytes'])
print(dataset.var()['dst_bytes'])
print(dataset.var()['land'])
print(dataset.var()['wrong_fragment'])
print(dataset.var()['urgent'])
print(dataset.var()['hot'])
print(dataset.var()['num_failed_logins'])
print(dataset.var()['logged_in'])
print(dataset.var()['lnum_compromised'])
print(dataset.var()['lroot_shell'])
print(dataset.var()['lsu_attempted'])
print(dataset.var()['lnum_root'])
print(dataset.var()['lnum_file_creations'])
print(dataset.var()['lnum_shells'])
print(dataset.var()['lnum_access_files'])
#print(dataset.var()['lnum_outbound_cmds'])
#print(dataset.var()['is_host_login'])
print(dataset.var()['is_guest_login'])
print(dataset.var()['count'])
print(dataset.var()['srv_count'])
print(dataset.var()['serror_rate'])
print(dataset.var()['srv_serror_rate'])
print(dataset.var()['rerror_rate'])
print(dataset.var()['srv_rerror_rate'])
print(dataset.var()['same_srv_rate'])
print(dataset.var()['diff_srv_rate'])
print(dataset.var()['srv_diff_host_rate'])
print(dataset.var()['dst_host_count'])
print(dataset.var()['dst_host_srv_count'])
```

```
print(dataset.var()['dst_host_same_srv_rate'])
print(dataset.var()['dst_host_diff_srv_rate'])
print(dataset.var()['dst_host_same_src_port_rate'])
print(dataset.var()['dst_host_srv_diff_host_rate'])
print(dataset.var()['dst_host_serror_rate'])
print(dataset.var()['dst_host_srv_serror_rate'])
print(dataset.var()['dst_host_rerror_rate'])
print(dataset.var()['dst_host_srv_rerror_rate'])
```

```
976576992036.6654
1091643891.0952706
4.453071700180602e-05
0.018172491590816044
3.03630038804738e-05
0.6116856844184144
0.00024085837553570224
0.12626868283082365
3.2339838746190672
0.0001113193556638224
6.072508173886537e-05
4.051043258019034
0.009296040477405787
0.000121440870502456
0.001330916397839991
0.001384663728080717
45431.6891034859
60674.89003505529
0.1449456310521699
0.14517386836860224
0.0536495355281265
0.05389232330783698
0.15069129958002264
0.006757755850526054
0.020276896258574477
4191.863370949552
11244.525010698932
0.1687402028545784
0.011937575833318366
0.2316583307814032
0.0017751826094751953
0.14485133584224127
0.1450998566016115
0.053171621563146344
0.052964669234579015
```

remove redundaant features

```
dataset['lsu_attempted'].value_counts()
dataset.drop('lsu_attempted', axis=1, inplace=True)
dataset['urgent'].value_counts()
dataset.drop('urgent', axis=1, inplace=True)
dataset['lnum_outbound_cmds'].value_counts()
dataset.drop('lnum_outbound_cmds', axis=1, inplace=True)
dataset['is_host_login'].value_counts()
dataset.drop('is_host_login', axis=1, inplace=True)
dataset['wrong_fragment'].value_counts()
```

```
dataset.drop('wrong_fragment', axis=1, inplace=True)
dataset['hot'].value_counts()
dataset.drop('hot', axis=1, inplace=True)
dataset['num_failed_logins'].value_counts()
dataset.drop('num_failed_logins', axis=1, inplace=True)
dataset['logged_in'].value_counts()
dataset.drop('logged_in', axis=1, inplace=True)
dataset['lroot_shell'].value_counts()
dataset.drop('lroot_shell', axis=1, inplace=True)
dataset['lnum_file_creations'].value_counts()
dataset.drop('lnum_file_creations', axis=1, inplace=True)
dataset['lnum_shells'].value_counts()
dataset.drop('lnum_shells', axis=1, inplace=True)
dataset['lnum_access_files'].value_counts()
dataset.drop('lnum_access_files', axis=1, inplace=True)
dataset['is_guest_login'].value_counts()
dataset.drop('is_guest_login', axis=1, inplace=True)
dataset['serror_rate'].value_counts()
dataset.drop('serror_rate', axis=1, inplace=True)
dataset['srv_serror_rate'].value_counts()
dataset.drop('srv_serror_rate', axis=1, inplace=True)


dataset['rerror_rate'].value_counts()
dataset.drop('rerror_rate', axis=1, inplace=True)
dataset['srv_rerror_rate'].value_counts()
dataset.drop('srv_rerror_rate', axis=1, inplace=True)
dataset['same_srv_rate'].value_counts()
dataset.drop('same_srv_rate', axis=1, inplace=True)
dataset['diff_srv_rate'].value_counts()
dataset.drop('diff_srv_rate', axis=1, inplace=True)
dataset['srv_diff_host_rate'].value_counts()
dataset.drop('srv_diff_host_rate', axis=1, inplace=True)
dataset['dst_host_same_srv_rate'].value_counts()
dataset.drop('dst_host_same_srv_rate', axis=1, inplace=True)
dataset['dst_host_diff_srv_rate'].value_counts()
dataset.drop('dst_host_diff_srv_rate', axis=1, inplace=True)
dataset['dst_host_same_src_port_rate'].value_counts()
dataset.drop('dst_host_same_src_port_rate', axis=1, inplace=True)
dataset['dst_host_srv_diff_host_rate'].value_counts()
dataset.drop('dst_host_srv_diff_host_rate', axis=1, inplace=True)
dataset['dst_host_serror_rate'].value_counts()
dataset.drop('dst_host_serror_rate', axis=1, inplace=True)
dataset['dst_host_srv_serror_rate'].value_counts()
dataset.drop('dst_host_srv_serror_rate', axis=1, inplace=True)
dataset['dst_host_rerror_rate'].value_counts()
dataset.drop('dst_host_rerror_rate', axis=1, inplace=True)
dataset['dst_host_srv_rerror_rate'].value_counts()
dataset.drop('dst_host_srv_rerror_rate', axis=1, inplace=True)


dataset.head()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	lnum_compromi
0	0	tcp	http	SF	181	5450	0	
1	0	tcp	http	SF	239	486	0	
2	0	tcp	http	SF	235	1337	0	
3	0	tcp	http	SF	219	1337	0	

```
map', 'perl', 'phf', 'pod', 'portsweep', 'rootkit', 'satan', 'smurf', 'spy', 'teardrop', '
```

```
dataset.describe()
```

	duration	src_bytes	dst_bytes	land	lnum_compromised	
count	494020.000000	4.940200e+05	4.940200e+05	494020.000000	494020.000000	494
mean	47.979400	3.025616e+03	8.685308e+02	0.000045	0.010212	
std	707.747185	9.882191e+05	3.304003e+04	0.006673	1.798328	
min	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	
25%	0.000000	4.500000e+01	0.000000e+00	0.000000	0.000000	
50%	0.000000	5.200000e+02	0.000000e+00	0.000000	0.000000	
75%	0.000000	1.032000e+03	0.000000e+00	0.000000	0.000000	
max	58329.000000	6.933756e+08	5.155468e+06	1.000000	884.000000	

```
x = dataset.iloc[:, :-1].values
#x
y = dataset.iloc[:, 13].values
#y
```

```
x
```

```
array([[0, 'tcp', 'http', ..., 8, 9, 9],
       [0, 'tcp', 'http', ..., 8, 19, 19],
       [0, 'tcp', 'http', ..., 8, 29, 29],
       ...,
       [0, 'tcp', 'http', ..., 18, 16, 255],
       [0, 'tcp', 'http', ..., 12, 26, 255],
       [0, 'tcp', 'http', ..., 35, 6, 255]], dtype=object)
```

```
x.shape
```

```
(494020, 13)
```

```
y.shape
```

```
(494020,)
```

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
labelencoder_x_1 = LabelEncoder()
labelencoder_x_2 = LabelEncoder()
labelencoder_x_3 = LabelEncoder()
x[:, 1] = labelencoder_x_1.fit_transform(x[:, 1])
x[:, 2] = labelencoder_x_2.fit_transform(x[:, 2])
x[:, 3] = labelencoder_x_3.fit_transform(x[:, 3])
```

```
x
```

```
array([[0, 1, 22, ..., 8, 9, 9],
       [0, 1, 22, ..., 8, 19, 19],
       [0, 1, 22, ..., 8, 29, 29],
       ...,
       [0, 1, 22, ..., 18, 16, 255],
       [0, 1, 22, ..., 12, 26, 255],
       [0, 1, 22, ..., 35, 6, 255]], dtype=object)
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state =
```

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
    verbose=False)
```

```
#Predicting the Test set results
y_pred = classifier.predict(X_test)
```

```
from sklearn import metrics
# Model Accuracy, how often is the classifier correct?
print(metrics.classification_report(y_test, y_pred))
```

```
                  precision    recall  f1-score   support
```

```
 #Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
 confusion_matrix(y_test, y_pred)
```

```
array([[117667,   1184],
       [   278,  29077]])
```

```
from sklearn.model_selection import GridSearchCV
parameters = [{'C': [1, 2], 'kernel': ['linear']},
              {'C': [1, 2], 'kernel': ['rbf'], 'gamma': [0.1, 0.2]}]
grid_search = GridSearchCV(estimator = classifier,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           cv = 2,
                           n_jobs = -1)
grid_search = grid_search.fit(X_train, y_train)
```

```
accuracy = grid_search.best_score_
```

```
accuracy*100
```

```
99.79526566304429
```