

```
from sklearn import datasets
from sklearn.feature_selection import VarianceThreshold
iris=datasets.load_iris()
X=iris.data
y=iris.target
thresholder=VarianceThreshold(threshold=.5)
X_high_variance=thresholder.fit_transform(X)
X_high_variance[0:5]

array([[5.1, 1.4, 0.2],
       [4.9, 1.4, 0.2],
       [4.7, 1.3, 0.2],
       [4.6, 1.5, 0.2],
       [5. , 1.4, 0.2]])

thresholder.fit(X).variances_




array([0.68112222, 0.18871289, 3.09550267, 0.57713289])

import pandas as pd
import numpy as np

X=np.array([[1,1,1],
           [2,2,0],
           [3,3,1],
           [4,4,0],
           [5,5,1],
           [6,6,0],
           [7,7,1],
           [8,7,0],
           [9,7,1]])

df=pd.DataFrame(X)

df
```

	0	1	2	
0	1	1	1	
1	2	2	0	
2	3	3	1	
3	4	4	0	
4	5	5	1	
5	6	6	0	
6	7	7	1	
7	8	7	0	
8	9	7	1	

Next steps:

[Generate code with df](#)[View recommended plots](#)



```

corr_matrix = df.corr().abs()
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
to_drop = [column for column in upper.columns if any(upper[column] > 0.95)]
df.drop(df[to_drop], axis=1)




```

	0	2	
0	1	1	
1	2	0	
2	3	1	
3	4	0	
4	5	1	
5	6	0	
6	7	1	
7	8	0	
8	9	1	

```
df.corr()
```

	0	1	2	
0	1.000000	0.976103	0.000000	
1	0.976103	1.000000	-0.034503	
2	0.000000	-0.034503	1.000000	

upper

	0	1	2	
0	NaN	0.976103	0.000000	
1	NaN	NaN	0.034503	
2	NaN	NaN	NaN	

Next steps:

[Generate code with upper](#)

☐ [View recommended plots](#)

```
df.drop(df[to_drop],axis=1)
```

	0	2	
0	1	1	
1	2	0	
2	3	1	
3	4	0	
4	5	1	
5	6	0	
6	7	1	
7	8	0	
8	9	1	

```
#Load libraries
from sklearn.datasets import load_iris
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

# Load iris data
iris=load_iris()

#Create features and target
X=iris.data
y=iris.target

#Convert to categorical data by converting data to integers
X = X.astype(int)

#Select two features with highest chi-squared statistics
from sklearn.datasets import load_iris
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif

iris=load_iris()
X=iris.data
y=iris.target

fvalue_selector=SelectKBest(f_classif,k=2)

X_kbest=fvalue_selector.fit_transform(X,y)

print('Original number of features:',X.shape[1])
print('Reduced number of features:',X_kbest.shape[1])

    Original number of features: 4
    Reduced number of features: 2
```