



```
import pandas as pd
import numpy as np

data = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
print(data)

# preg - pregnancy
# plas = plasma
# pres = pressure
features = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

df = pd.read_csv(data, names=features)
df.head() # class 1 represents person having diabetes and class 0 means person not having diabetes
```

 <https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv>

	preg	plas	pres	skin	test	mass	pedi	age	class	
0	6	148	72	35	0	33.6	0.627	50	1	
1	1	85	66	29	0	26.6	0.351	31	0	
2	8	183	64	0	0	23.3	0.672	32	1	
3	1	89	66	23	94	28.1	0.167	21	0	
4	0	137	40	35	168	43.1	2.288	33	1	

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

Start coding or [generate](#) with AI.

```
# printing the shape
df.shape
```

 (768, 9)

✓ Preparing the Data


```
data = df.values
X = data[:, 0:8]
Y = data[:, 8]
```

✓ Filter Method

```
# importing the required libraries
from sklearn.feature_selection import SelectKBest # importing the Select K-Best
from sklearn.feature_selection import chi2 # importing chi-square test
```

```
# feature selection
chi_best = SelectKBest(score_func=chi2, k = 4)
print("----- Chi best ----- ")
print(chi_best)
k_best = chi_best.fit(X,Y)
print()
# summarize the scores
np.set_printoptions(precision=3)
print("---- K best scores-----")
print(k_best.scores_)

print()
k_features = k_best.transform(X)
# Summarize the selected features
print("----- Selected Features k features top 5 features -----")
print(k_features[0:5, :])
```

 ----- Chi best -----
 SelectKBest(k=4, score_func=<function chi2 at 0x7a8d11fc4280>)

---- K best scores-----
 [111.52 1411.887 17.605 53.108 2175.565 127.669 5.393 181.304]

```

----- Selected Features k features top 5 features -----
[[148.    0.   33.6  50. ]
 [ 85.    0.   26.6  31. ]
 [183.    0.   23.3  32. ]
 [ 89.   94.   28.1  21. ]
 [137.  168.   43.1  33. ]]

```

✓ Wrapper Method

Recursive Feature Elimination

```

from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

```

```

import warnings
warnings.filterwarnings('ignore')

```

```

data = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
print(data)

```

```

# preg - pregnancy
# plas = plasma
# pres = pressure
features = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']

```

```

df = pd.read_csv(data, names=features)
df.head() # class 1 represents person having diabetes and class 0 means person not having diabetes

```

```

data = df.values
X = data[:, 0:8]
Y = data[:, 8]

```

```

# feature extraction
model_lr = LogisticRegression()
recur_fe = RFE(model_lr, n_features_to_select=4)
Feature = recur_fe.fit(X, Y)
print("Number of Features %s" %(Feature.n_features_))
print("Selected Features %s" %(Feature.support_))
print("Feature Ranking %s" %(Feature.ranking_))

```

```

➦ https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv
Number of Features 4
Selected Features [ True  True False False False  True  True False]
Feature Ranking [1 1 3 4 5 1 1 2]

```

✓ Embedded Method Techniques

Ridge Regression / L2 regularization that doesnot make the absoulte weight 0

```

from sklearn.linear_model import Ridge

```

```

ridge_reg = Ridge(alpha=0.01)
ridge_reg.fit(X,Y)

```

```

➦
▼ Ridge ⓘ ?
Ridge(alpha=0.01)

```

```

# Helper Function for printing the coefficients

```

```

def print_coefs(coef, names=None, sort=False):
    if names == None:
        names = ['X%s' % x for x in range(len(coef))]
    list = zip(coef, names)
    if sort:
        list = sorted(list, key=lambda x: -abs(x[0]))
    return " + ".join("%s * %s" % (round(coefs, 3), name) for coefs, name in list)

```

```

print("Ridge model: ", print_coefs(ridge_reg.coef_))

```

 Ridge model: $0.021 * X_0 + 0.006 * X_1 + -0.002 * X_2 + 0.0 * X_3 + -0.0 * X_4 + 0.013 * X_5 + 0.147 * X_6 + 0.003 * X_7$