```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving iris.csv to iris.csv
```

```
import csv
import math

def read_iris_csv(filename):
    data = []
    with open(filename, 'r') as f:
        for row in csv.reader(f):
            data.append(list(map(float,row[:4])))
    return data

def mean(data):
    n = len(data)
    m = len(data[0])
    means = [0]*m
    for r in data:
        for i in range(m):
            means[i]+=r[i]
    for i in range(m):
        means[i]/=n
    return means

def center(data, means):
    return [[r[i]-means[i] for i in range(len(r))] for r in data]

def covariance_matrix(data):
    n = len(data)
    m = len(data[0])
    cov = [[0]*m for _ in range(m)]
    for i in range(m):
        for j in range(m):
            s=0
            for k in range(n):
                s+=data[k][i]*data[k][j]
            cov[i][j]=s/(n-1)
    return cov

def mat_vec_mult(mat, vec):
    return [sum(mat[i][j]*vec[j] for j in range(len(vec))) for i in range(len(mat))]

def vec_norm(vec):
    return math.sqrt(sum(x*x for x in vec))

def normalize(vec):
    norm = vec_norm(vec)
    return [x/norm for x in vec]

def power_iteration(mat, iterations=1000):
    n = len(mat)
    b = [1]*n
    for _ in range(iterations):
        b = normalize(mat_vec_mult(mat,b))
    Ab = mat_vec_mult(mat,b)
    eigenvalue = sum(Ab[i]*b[i] for i in range(n))
    return eigenvalue,b

def deflate(mat, val, vec):v
    n = len(mat)
    for i in range(n):
        for j in range(n):
            mat[i][j] -= val*vec[i]*vec[j]
    return mat

def pca(data, k=2):
    means = mean(data)
    centered = center(data,means)
    cov = covariance_matrix(centered)
    cov_copy = [row[:] for row in cov]
    eigenvalues = []
    eigenvectors = []
    for _ in range(k):
        val, vec = power_iteration(cov_copy)
        eigenvalues.append(val)
        eigenvectors.append(vec)
        cov_copy = deflate(cov_copy,val,vec)
    projected = []
    for r in centered:
        projected.append([sum(r[i]*eigenvectors[j][i] for i in range(len(r))) for j in range(k)])
    return projected,eigenvalues,eigenvectors

data = read_iris_csv('iris.csv')
proj,eigvals,eigvecs = pca(data,2)
print("Eigenvalues:", eigvals)
print("Eigenvectors:", eigvecs)
print("Projected data (first 5 rows):", proj[:5])
```

```
Eigenvalues: [0.16657863132827852, 0.020655094754129924]
Eigenvectors: [[0.668111019795272, 0.7166344774801544, 0.164411275166307, 0.11415613655453069], [-0.7064764857539528,
Projected data (first 5 rows): [[0.28600350898268906, -0.04691971512801874], [-0.20593593371644206, -0.226976224751361
```

machine learning in python