* PCA过程：
* 1.特征中心化。即每一维的数据都减去该维的均值。这里的“维”指的就是一个特征（或属性），得到矩阵B
* 2. 计算B的协方差矩阵C
* 3.计算协方差矩阵C的特征值和特征向量
* 4.选取大的特征值对应的特征向量，得到新的数据集
* C : float, optional (default=1.0) 误差项的惩罚参数，一般取值为10的n次幂，如10的-5次幂，10的-4次幂。。。。10的0次幂，10，1000,1000，在python中可以使用pow（10，n） n=-5~inf C越大，相当于惩罚松弛变量，希望松弛变量接近0，即对误分类的惩罚增大，趋向于对训练集全分对的情况，这样会出现训练集测试时准确率很高，但泛化能力弱。 C值小，对误分类的惩罚减小，容错能力增强，泛化能力较强。
* kernel : string, optional (default=’rbf’) svc中指定的kernel类型。 可以是： ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’ 或者自己指定。 默认使用‘rbf’ 。
* degree : int, optional (default=3) 当指定kernel为 ‘poly’时，表示选择的多项式的最高次数，默认为三次多项式。 若指定kernel不是‘poly’,则忽略，即该参数只对‘poly’有作用。
* gamma : float, optional (default=’auto’) 当kernel为‘rbf’, ‘poly’或‘sigmoid’时的kernel系数。 如果不设置，默认为 ‘auto’ ，此时，kernel系数设置为：1/n\_features
* coef0 : float, optional (default=0.0) kernel函数的常数项。 只有在 kernel为‘poly’或‘sigmoid’时有效，默认为0。
* probability : boolean, optional (default=False) 是否采用概率估计。 必须在fit（）方法前使用，该方法的使用会降低运算速度，默认为False。
* shrinking : boolean, optional (default=True) 如果能预知哪些变量对应着支持向量，则只要在这些样本上训练就够了，其他样本可不予考虑，这不影响训练结果，但降低了问题的规模并有助于迅速求解。进一步，如果能预知哪些变量在边界上(即a=C)，则这些变量可保持不动，只对其他变量进行优化，从而使问题的规模更小，训练时间大大降低。这就是Shrinking技术。 Shrinking技术基于这样一个事实：支持向量只占训练样本的少部分，并且大多数支持向量的拉格朗日乘子等于C。
* tol : float, optional (default=1e-3) 误差项达到指定值时则停止训练，默认为1e-3，即0.001。
* cache\_size : float, optional 指定内核缓存的大小，默认为200M。
* class\_weight : {dict, ‘balanced’}, optional 权重设置。如果不设置，则默认所有类权重值相同。 以字典形式传入。
* verbose : bool, default: False 是否启用详细输出。 多线程时可能不会如预期的那样工作。默认为False
* max\_iter : int, optional (default=-1) 强制设置最大迭代次数。 默认设置为-1，表示无穷大迭代次数。 Hard limit on iterations within solver, or -1 for no limit
* random\_state : int, RandomState instance or None, optional (default=None) 伪随机数使用数据。