特征选择：

1.Pearson 相关系数

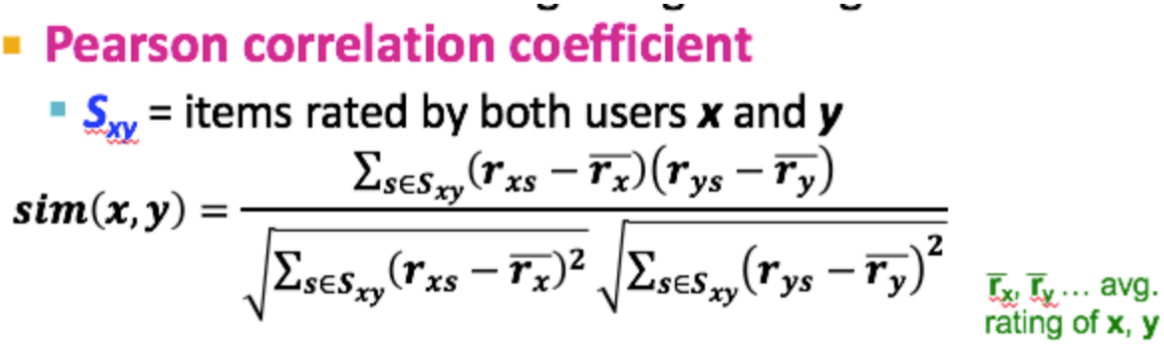
衡量的是变量之间的线性相关性，结果的取值区间为**[-1,1]**， -1 表示完全的负相关(这个变量下降，那个就会上升)， +1 表示完全的正相关， 0 表示没有线性相关性。

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y},$$

Pearson相关系数的一个**明显缺陷**是，作为特征排序机制，他**只对线性关系敏感**。如果关系是非线性的，即便两个变量具有一一对应的关系，Pearson相关性也可能会接近 0 。

（用数学推导举一个例子）

然后问题来了, 皮尔逊系数和这个cos啥关系...(不好意思借用了我们学校老师的课件...)



皮尔森相关系数计算公式

其实皮尔逊系数就是cos计算之前两个向量都先进行中心化(centered)...就这么简单...

1.简单的例子

```
x  = np.array([1,2,3,4,5])
y = np.array([4,5,6,7,8])
pearsonr(x,y)
```

2.复杂的例子

```
[pearsonr(np.array(X.iloc[:,i]),Y) for i in range(len(X.T))]
```

3真实场景应用：

```
[x for x in X.T]  #[1,2,3,4,5...]
#选取每一行和每一列的所有数据
[x for x in X.values.T]  #[array([6.32000e-03, 2.73100e-02, 2.72900e-02, 3.23700e-02,
6.90500e-02,..]
```

```
func_pearson = lambda X,y: list (np.array ([pearsonr (x,y) for x in X.T]).T)
X_filtered = SelectKBest(func_pearson, k = 5).fit_transform(X.values, Y)
X_filtered.shape, X.shape

skb=SelectKBest(func_pearson, k = 5)
skb.fit(X, Y)
skb.get_support(indices=True)
#哪几列?
```

2.卡方检验

经典的卡方检验是检验**类别型变量**对**类别型变量**的相关性。假设自变量有N种取值，因变量有M种取值，考虑自变量等于i且因变量等于j的样本频数的观察值与期望的差距，构建统计量：

$$\chi^2 = \sum \frac{(A-E)^2}{E}$$

```
from sklearn.datasets import load_iris
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
iris = load_iris()
X, y = iris.data, iris.target  #iris数据集

#选择K个最好的特征，返回选择特征后的数据
X_new = SelectKBest(chi2, k=2).fit_transform(X, y)
```

卡方值越大，对结果影响就越大

3.VarianceThreshold 方差选择法

使用方差选择法，先要计算各个特征的方差，然后根据阈值选择大于该阈值的feature

```
from sklearn.feature_selection import VarianceThreshold
#方差选择法，返回值为特征选择后的数据 #参数threshold为方差的阈值
from sklearn.datasets import load_iris
iris = load_iris()
selector = VarianceThreshold(threshold=3).fit(iris.data, iris.target)
data = selector.transform(iris.data)
print(data[0:5])
print(selector.variances_)
```

4.互信息和最大信息系数 Mutual information and maximal information coefficient (MIC)

MIC处理非线性的数据更有效

想把互信息直接用于特征选择其实不是太方便：

1.它不属于度量方式，也没有办法归一化，在不同数据及上的结果无法做比较

2.对于连续变量的计算不是很方便（ X 和 Y 都是集合, x_i, y 都是离散的取值），通常变量需要先离散化，而互信息的结果对离散化的方式很敏感

最大信息系数克服了这两个问题。它首先寻找一种最优的离散化方式，然后把互信息取值转换成一种度量方式，取值区间在 [0,1] 。

互信息(Mutual Information)：

是信息论里一种有用的信息度量，它可以看成是一个随机变量中包含的关于另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不肯定性。

系统学习特征工程吧；

依我个人经验，特征工程的"醍醐灌顶"其实更依赖于domain knowledge

但是拥有完整的特征工程理论知识，是应用domain knowledge的基础

```
import pandas as pd

fi = pd.DataFrame({'feature': list(train.columns),
'importance': model.feature*importances*}).
sort_values('importance', ascending = False)

fi.head()
```