# فهرست مطالب

[فهرست مطالب 1](#_Toc27765711)

[مقدمه 2](#_Toc27765712)

[زبان برنامه نویسی استفاده شده 2](#_Toc27765713)

[پکیج‌های استفاده شده 3](#_Toc27765714)

[خواندن دیتاست 3](#_Toc27765715)

[کاهش میانگین از داده‌ها 3](#_Toc27765716)

[محاسبه تجزیه مقادیر منفرد (SVD) 3](#_Toc27765717)

[محاسبه مقادیر ویژه و بردار ویژه 3](#_Toc27765718)

[محاسبه واریانس تجمعی و بدست آوردن عدد k 4](#_Toc27765719)

[رسم نمودار واریانس تجمعی 5](#_Toc27765720)

[محاسبه ماتریس Score و loading 6](#_Toc27765721)

گزارش مربوط به پروژه پایانی درس داده کاوی محاسباتی ترم پاییز سال 1398 قسمت اول

نام اساتید: دکتر نریمانی و دکتر وثیقی

نام دانشجویان: کامران عبدی (984101) و رضا خالقی (984118)

# مقدمه

مهم‌ترین هدف، آنالیز مولفه اصلی (PCA) تحلیل داده‌ها، جهت پیدا کردن یک الگو، برای کاهش ابعاد است، بطوری که با کمتری میزان از دست دادن اطلاعات همراه باشد.

در بخش اول این پروژه ما قصد داریم مولفه‌های اصلی داده‌های بیان ژن که مربوط به سرطان پستان می‌باشد، استخراج کنیم. تا دیتاست خود را از یک فضای ویژگی با ابعاد بزرگ به یک فضای ویژگی کوچکتری که داده‌هایمان را به "بهترین" نحو توصیف می‌کند، نگاشت کنیم. تا از این طریق هزینه‌های محاسباتی و میزان خطای خود را کاهش دهیم.

این مجموعه داده از اندازه‌گیری بیان 47293 ژن در نمونه‌های 128 نفر بدست آمده است. افراد مورد مطالعه در دو دسته luminal و non-luminal قرار دارند.

بنابراین هدف پیدا کردن یک k است که ابعاد داده‌های ما را از 47293 بعد به k بعد کاهش دهیم.

مراحل انجام پروژه:

1. خواندن دیتاست از فایل اکسلی بنام breast\_preprocessed.xls که در مسیر پروژه و در فولدر جداگانه‌ای بنام folders قرار دارد.
2. کاهش میانگین از داده‌های بیان ژن
3. انجام تجزیه SVD
   1. محاسبه مقدار ویژه، بردار ویژه
   2. مرتب سازی مقدار ویژه از بزرگ به کوچک
   3. گرفتن ریشه دوم مقادیر ویژه و تبدیل به یک ماتریس قطری
4. محاسبه میزان واریانس تجمعی و بدست آوردن k بطوری که با کمترین میزان از دست دادن اطلاعات همراه باشد. (در این پروژه ما k رو طوری بدست آوردیم که بتوانیم 90 درصد داده ها را حفظ کنیم)
5. رسم واریانس تجمعی
6. محاسبه ماتریس Score
7. رسم ماتریس loading و رسم آن

# زبان برنامه نویسی استفاده شده

پایتون 3.7

# پکیج‌های استفاده شده

* pandas
* numpy
* matplotlib.pyplot
* os

# خواندن دیتاست

از طریق تابع read\_data نوشته شده توسط ما، فایل اکسل بنام breast\_preprocessed.xls که در مسیر پروژه و فولدری بنام files قرار دارد خوانده می‌شود. نوع داده‌ای فایل خوانده شده DataFrame می‌باشد، که شامل داده‌های بیان ژن و نیز برچسب داده‌ها و اسامی ژن‌هاست.

data = pd.read\_excel(filepath, header=**None**)

داده‌های فوق را که داخل متغیری بنام data ذخیره شده است از طریق تابع array پکیج numpy تبدیل به ماتریس می‌نماییم و دو سطر انتهای آن که مربوط به عنوان کلاس داده‌هاست در نظر نمی‌گیریم و نیز از طریق تابع transpose() ماتریس را می چرخانیم. یعنی نمونه ها در سطر ماتریس و متغیرها در ستون ماتریس قرار می‌گیرند:

X = np.array(data.iloc[0:len(data.index)-2, 1:], dtype = np.float64).transpose()

# کاهش میانگین از داده‌ها

تابع mean\_centered

میانگین سطر به سطر ماتریس (میانگین اطلاعات هر ژن) محاسبه می‌شود:

mean\_vector = np.mean(X, axis = 0)

کاهش میانگین بصورت سطر به سطر:

A = A - mean\_vector

بنابراین اکنون از ماتریس داده‌های ما میانگین داده‌ها کم شده است و به اصطلاح تبدیل به دیتای Mean centered شده است.

# محاسبه تجزیه مقادیر منفرد (SVD)

بطور خلاصه می‌توان گفت که تجزیه مقادیر منفرد روشی است که یک ماتریس به سه ماتریس دیگر تجزیه می‌کند.

A = USVt

## محاسبه مقادیر ویژه و بردار ویژه

بدلیل اینکه تعداد سطرها کمتر از تعداد ستون‌هاست محاسبه A.dot(A.T) ساده تر از محاسبه ماتریس A.T.dot(A) است:

eig\_vals, eig\_vecs = np.linalg.eig(A.dot(A.T))

از طریق کد بالا مقادیر ویژه و بردارهای ویژه از ماتریس A.At استخراج می‌شود.

مقادیر ویژه بصورت نزولی مرتب می‌شود:

sorted\_eigen\_vals = sorted (eig\_vals, reverse=**True**)

در مقادیر ویژه یک عدد منفی (-0.00003) وجود داشت. که با مرتب سازی داده از بزرگ به کوچک به انتهای ساختار داده منتقل شد.

*نکته) در ابتدا در اثر اشتباه در خواندن داده ها سطر اول فایل اکسل را نمی خواندیم که بردار ویژه مورد اشاره در بالا هم ایجاد نمی‌شد ولی پس از متوجه شدن اشتباه رخ داده و خواندن سطر اول فایل اکسل یک عدد منفی در لیست مقدار ویژه پیدا شد. البته یک تغییر دیگر نیز برای ignore شدن این مقدار ویژه انجام شد. نوع داده‌ای ماتریس از float32 به float64 تغییر دادیم و متوجه شدیم که آن عدد مقدار ویژه به صفر تبدیل شد و عملا پس از پیدا کردن عدد k و خواندن k مقدار ویژه ابتدایی از محاسبات ما بکلی حذف شد.*

ماتریس سیگما همان ریشه دوم مقادیر ویژه است که بصورت قطری در کنار هم چیده می‌شود و سایر مقادیر ماتریس صفر است.

S = np.diag(np.sqrt(sorted\_eigen\_vals))

هنگام توسعه کد به دلیل اینکه بصورت مکرر لازم بود که فایل اکسل خوانده شود تعداد 1500 سطر ابتدایی انتخاب و بقیه را حذف کرده بودیم، در آن فایل تستی موقتی، آن مقدار ویژه منفی که در بالا به آن اشاره شد عددی بزرگتر (قدرمطلق) بود که پس از گرفتن ریشه دوم مقادیر ویژه مقدار nan در پایتون تولید می‌شد بنابراین از طریق کد زیر مقدار nan در ماتریس مقادیر ویژه به صفر تبدیل شد:

S = np.nan\_to\_num(S)

نکته: در تست‌ها نهایی از همان فایل اصلی با همان تعداد سطر داده‌ها استفاده شد.

تعدادی عناصر قطر اصلی ماتریس سیگما:

1.81879152e+02, 1.65501312e+02, 1.53158612e+02

# محاسبه واریانس تجمعی و بدست آوردن عدد k

تابع cumulative\_var

محاسبه مجموع قطر ماتریس سیگما:

sTot = np.nansum(S)

از طریق کد زیر حاصل جمع هر عدد هر خانه ماتریس S (سیگما در SVD) تا اولین عدد تقسیم بر مجموع مقادیر ویژه ضربدر 100 محاسبه شده و در ماتریسی بنام var\_i ذخیره می‌شود.

var\_i = np.array([(np.sum(S[: i + 1]) / sTot) \* 100.0 **for** i **in** range(row)])

row: بعد سطر‌ها یعنی 128

sTot: مجموع مربع مقادیر ویژه که همان عناصر قطر اصلی ماتریس سیگما می‌باشد.

سپس عدد k (حداقل تعداد ویژگی‌هایی که باید نگهداشته شود تا 90 درصد اطلاعات داده‌ها حفظ شود) را از طریق کد زیر محاسبه کردیم:

k = len(var\_i[var\_i < (varRetained \* 100)])

varRetained: 0.9

مقدار k عدد 106 بدست آمد و عنصر خانه 106 ام ماتریس var\_i عدد 90.20% می‌باشد.

تعدادی از عناصر ماتریس واریانس تجمعی:

[ 3.11394604 5.94748771 8.56971048 10.64924263 12.42461895

14.15960482 15.77624814 17.2193634 18.58551577 19.88017961

21.11978834 22.31659569 23.46717769 24.58376612 25.63478864

...

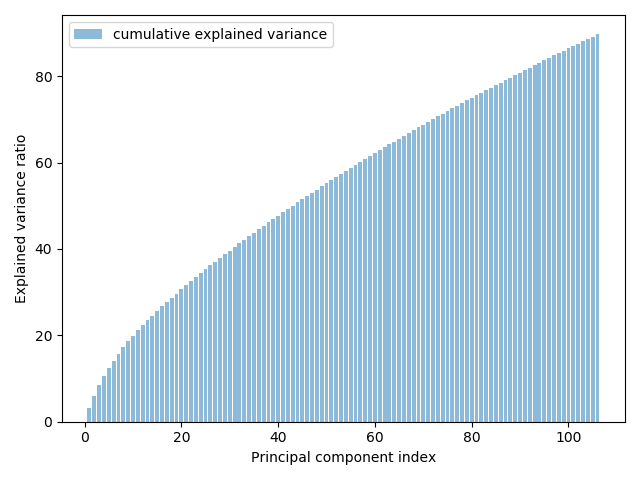
]

# رسم نمودار واریانس تجمعی

تابع cumulative\_variance\_plot را جهت رسم نمودار ستونی ماتریس واریانس تجمعی ایجاد کرده‌ایم.

Plt.bar(range(1, k + 1), var\_i[:k], alpha=0.5, align=**’center’**, label=**’cumulative explained variance’**)  
plt.ylabel(**‘Explained variance ratio’**)  
plt.xlabel(**‘Principal component index’**)  
plt.legend(loc=**’best’**)  
plt.show()

خروجی تابع فوق نمودار زیر می‌باشد:



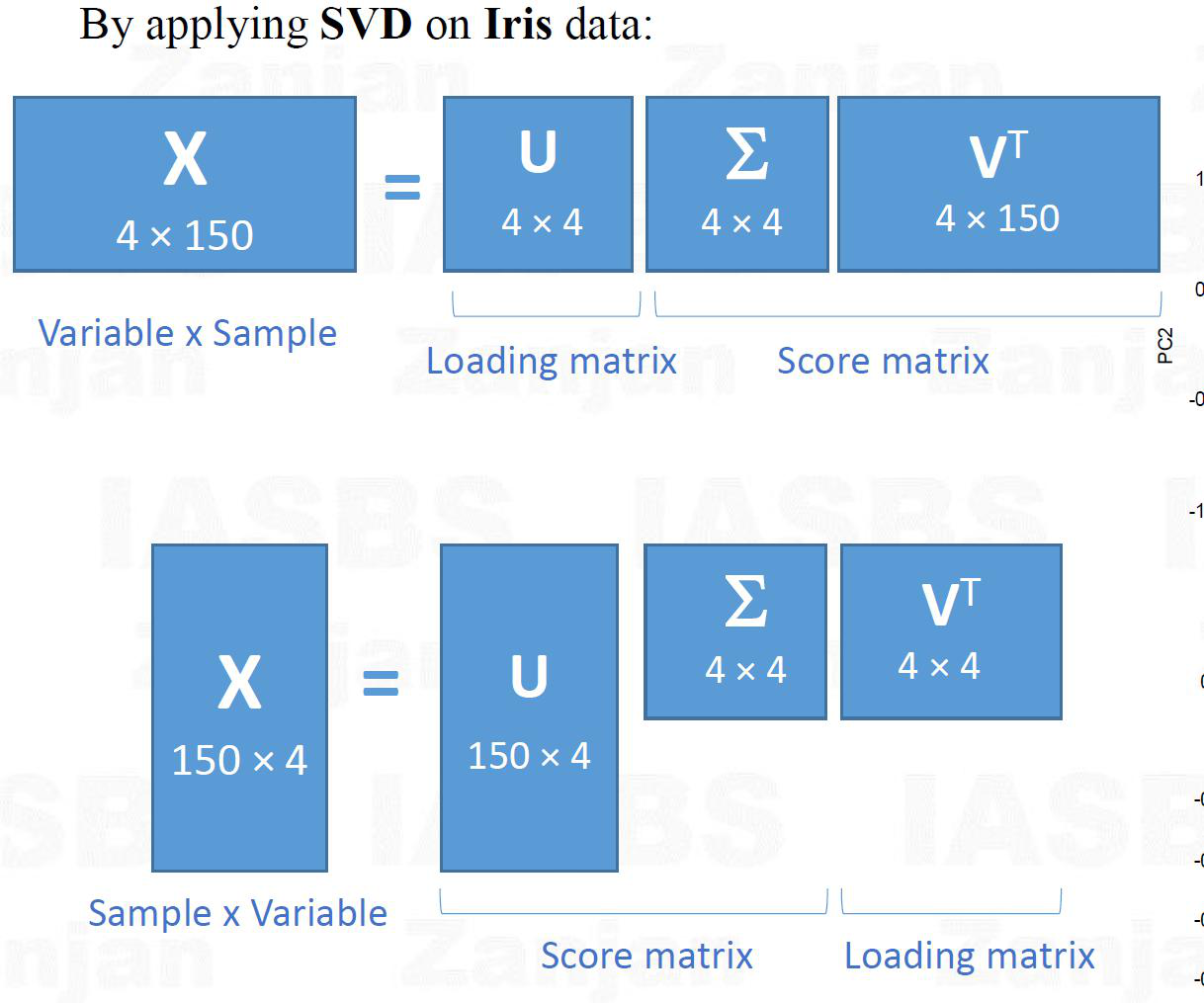
نمودار فوق نشان می‌دهد که 128 ویژگی شامل 100 درصد اطلاعات اکسل می‌باشد و ما تعداد 106 ویژگی را جهت حفظ بیش از 90 درصد اطلاعات حفظ می‌کنیم و بقیه ویژگی‌ها در جهت کاهش ابعاد و سهولت محاسبات و کاهش میزان خطا حذف می‌شود.

# محاسبه ماتریس Score و loading

لیست توابع نوشته شده جهت انجام این محاسبات:

* score\_matrix\_plot
* calc\_loading\_matrix
* loading\_matrix\_plot

طبق شکل صفحه 19 اسلاید شماره 14 درس داده‌کاوی محاسباتی روش محاسبه Score بصورت زیر می‌باشد:



Score\_matrix = U.dot(S)

ماتریس فوق یک ماتریس مربعی 128 در 128 می‌باشد.

و از طریق کد زیر نمودار score matrix رسم می‌شود:

plt.scatter(Score\_matrix[:, 0], Score\_matrix[:, 1])

loading = Vt

X = USVt

# *calculate VT matrix*

U\_reduced = U[:, : k]  
S\_reduced = S[: k, : k]  
S\_inv = np.linalg.inv(S\_reduced)  
VT = S\_inv.dot(U\_reduced.T).dot(X)  
loading\_matrix = VT