Singular Value Decomposition:

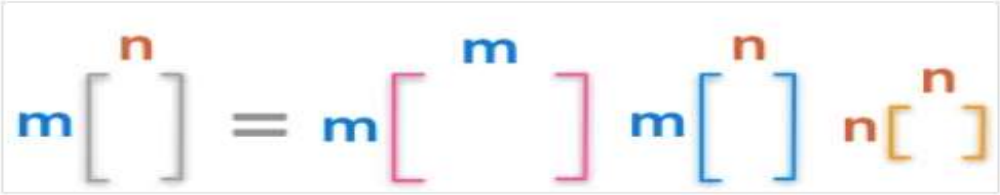
(SVD) یک ماتریس، فاکتورسازی آن ماتریس به سه ماتریس است. این ویژگی های جبری جالبی دارد و بینش های هندسی و نظری مهمی را در مورد تبدیل های خطی منتقل می کند. همچنین کاربردهای مهمی در علم داده دارد. SVD یک تکنیک پرکاربرد برای تجزیه یک ماتریس به چندین ماتریس جزء است که بسیاری از خواص مفید و جالب ماتریس اصلی را نشان می دهد. با استفاده از SVD، می‌توانیم رتبه ماتریس را تعیین کنیم، حساسیت یک سیستم خطی به خطای عددی را کمی کنیم، یا یک تقریب بهینه رتبه پایین‌تر به ماتریس به دست آوریم.

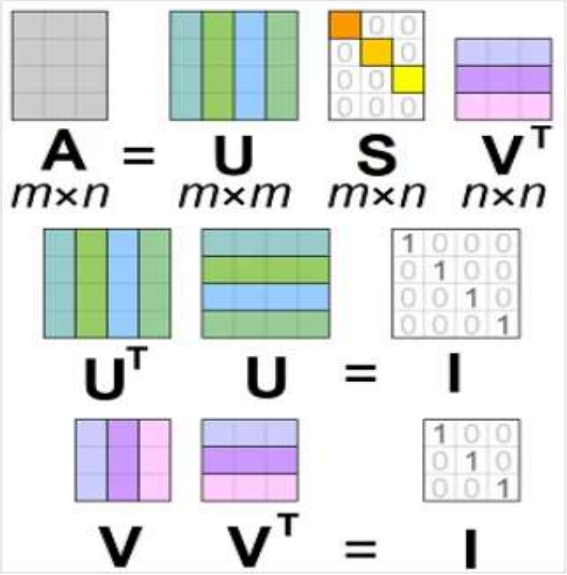
محاسبات تجزیه مقادیر منفرد:

$$A = USV^T$$

که در آن:

- Aیک ماتریس m×n ؛
- Uیک ماتریس متعامد m×m ؛
- Sیک ماتریس قطری m×n
- و Vیک ماتریس متعامد n×n است.





یکی چالش‌های متداول در یادگیری ماشین، وجود چند صد متغیر است، در حالی که بسیاری از الگوریتم‌های یادگیری ماشین، اگر با تعدادی بیش از یک مقدار مشخص متغیر کار کنند، با شکست مواجه می‌شوند. این موضوع، استفاده از تجزیه مقادیر تکین را برای کاهش متغیر در یادگیری ماشین ضروری می‌کند.

اگر تعداد و انواع مناسبی از ویژگی‌ها برای حل یک مساله خاص به الگوریتم‌های یادگیری ماشین داده شود، این الگوریتم‌ها به خوبی کار می‌کنند. اما، در صورتی که تعداد ویژگی‌ها (متغیرها) بسیار زیاد باشد، اغلب الگوریتم‌های یادگیری ماشین در حل مسئله دچار مشکل می‌شوند، زیرا با مسئله (High Dimensional Data) مواجه خواهیم بود. در اینجا است که بحث کاهش ابعاد (Dimensionality Reduction)مطرح می‌شود.

SVD با کاهش تعداد مقادیر تکین می‌تواند یک ماتریس را بسیار دقیق تقریب بزند. از این ویژگی می‌توان برای فشرده‌سازی داده‌ها با فرم‌های کوتاه شده U، Sو Vبه جای Aو برای کاهش متغیر از جایگزینی Aبا Uاستفاده کرد.

در حل بسیاری از مسائل، مشکلاتی وجود دارد که نیازمند روش‌هایی برای حل آن‌ها می‌باشیم. تجزیه مقدار تکین یکی از این روش‌ها می‌باشد که:

۱. در فضای پر داده و دارای خطاهایی نظیر خطای گردکردن قابل استفاده می‌باشد.
۲. در حل معادلات و ماتریس‌ها نزدیک به تکینگی و در حالت تکینه قابل استفاده است.
۳. می‌تواند مشکل حل را به‌طور دقیق باز کند.
۴. گاهی می‌تواند علاوه بر شرح مشکل مسئله آن را نیز حل کند.

الگوریتم KNN:

الگوریتم کی-نزدیکترین همسایه یک متد آمار ناپارامتری است که برای طبقه‌بندی آماری و رگرسیون استفاده می شود. در هر دو حالت کی شامل نزدیک ترین مثال آموزشی در فضای داده ای می باشد و خروجی آن بسته به نوع مورد استفاده در طبقه بندی و رگرسیون متغیر است. در حالت طبقه بندی با توجه به مقدار مشخص شده برای k، به محاسبه فاصله نقطه ای که میخواهیم برچسب آن را مشخص کنیم با نزدیک ترین نقاط میپردازد و با توجه به تعداد رای حداکثری این نقاط همسایه، در رابطه با برچسب نقطه مورد نظر تصمیم گیری می‌کنیم. برای محاسبه این فاصله میتوان از روش های مختلفی استفاده کرد که یکی از مطرح ترین این روش ها، فاصله اقلیدسی است. در حالت رگرسیون نیز میانگین مقادیر بدست آمده از k خروجی آن می باشد. از آنجا که محاسبات این الگوریتم بر اساس فاصله است نرمال‌سازی داده‌ها می‌تواند به بهبود عملکرد آن کمک کند.

مراحل الگوریتم knn شامل موارد زیر خواهد بود:

۱. بارگیری داده ها.

۲. Kبه عنوان تعداد نزدیک‌ترین همسایگان انتخاب می شود.
۳. برای هر یک از داده‌های اولیه:
۱. فاصله بین داده مورد سؤال و هر یک داده‌های اولیه را محاسبه می شود.
۲. فاصله و اندیس نمونه را به یک مجموعه اضافه میکنیم.
۳. مجموعه را بر اساس فاصله از کوچک به بزرگ مرتب میکنیم.
۴. نقاط K عضو اول مجموعه مرتب شده را انتخاب میکنیم.
۵. بسته به حالت یا حالت طبقه‌بندی، خروجی را اعلام میکنیم.

اگر مجموعه داده های ما دارای اندازه بزرگ باشد از SVD استفاده میکنیم و در حالتی که مجموعه داده هایی با نسبت داده های گمشده کم داشته باشیم KNN بهتر عمل میکند.یکی دیگر از تفاوت ها این است که الگوریتم KNN بر روی داده های خام بر خلاف SVD عمل می کند.

نصب surprise:

-برای نصب Surprise ابتدا anaconda یا miniconda را نصب میکنیم و سپس با دستور زیر می توان Surprise را نصب کرد:

```
conda install -c conda-forge scikit-surprise
```

تابع cross_validate() :

تابع validate_cross() یک Validation Cross را با توجه به آرگومان cv(4) با توجه به سوال اجرا می کند و با توجه به معیارهای مشخص شده MAE و RMSE دقت را محاسبه می کند.

RMSE:

تفاوت میان مقدار پیش‌بینی شده توسط مدل یا برآوردگر آماری و مقدار واقعی می‌باشد RMSE .میزان خطای بین دو مجموعه داده را اندازه گیری میکند. این پارامتر معمولاً مقادیر پیش بینی شده و مقادیر اندازه گیری شده را با یکدیگر مقایسه میکند.

MAE:

میانگین قدرمطلق خطا (Mean Absolute Error) است که به اختصار MAE نیز نامیده می‌شود. این تابع زیان، به مانند MSEاز فاصله بین مقدار پیش‌بینی و واقعی به عنوان معیار استفاده کرده ولی جهت این تفاضل را در نظر نمی‌گیرد. بنابراین در محاسبه خطا MAEفقط میزان فاصله و نه جهت فاصله به کار می‌رود.

بنابراینMAE ، میانگین قدرمطلق تفاضل بین مقدار پیش‌بینی و واقعی را محاسبه می‌کند. شیوه بدست آوردن MAE در رابطه زیر نوشته شده است.

$$MAE = \frac{\sum |y_i - \hat{y}_i|}{n}$$

اگر داده‌های پرت، حاصل مشاهدات مخدوش باشد، بهتر است از MAE استفاده کنیم تا اثر آن‌ها را در برآورد پارامترهای مدل از بین ببریم.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

```
⋯   Evaluating RMSE, MAE of algorithm SVD on 4 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Mean    Std
RMSE (testset)    0.9405  0.9453  0.9361  0.9358  0.9394  0.0039
MAE (testset)     0.7414  0.7450  0.7387  0.7395  0.7412  0.0024
Fit time          8.49    7.65    8.09    9.04    8.32    0.51
Test time         0.39    0.41    0.67    0.32    0.44    0.13
```

```
Evaluating RMSE, MAE of algorithm KNNBasic on 4 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Mean    Std
RMSE (testset)    0.9877  0.9775  0.9743  0.9878  0.9818  0.0060
MAE (testset)     0.7815  0.7728  0.7687  0.7787  0.7754  0.0050
Fit time          0.45    0.60    0.59    0.60    0.56    0.06
Test time         4.13    5.02    4.87    4.86    4.72    0.35
```

:۸

Uid=225

Iid=306

SVD:

```
user: 225        item: 306        r_ui = 4.00   est = 4.29   {'was_impossible': False}
```

KNN:

```
user: 225        item: 306        r_ui = 4.00   est = 4.13   {'actual_k': 40, 'was_impossible': False}
```

Uid=505

Iid=101

SVD:

```
user: 505        item: 101        r_ui = 4.00   est = 3.15   {'was_impossible': False}
```

KNN:

```
user: 505        item: 101        r_ui = 4.00   est = 3.28   {'actual_k': 40, 'was_impossible': False}
```