Trabalho – Análise PCA

Allan Rodrigues Rebelo
Bruno Abreu Kemmer Nº 5910474
Luciana Silva Souza Nº 9031317

Introdução

Após analisarmos o dataset atual (base de startups retirada do site crunchbase de 2013), verificamos que a maioria de suas variáveis eram categóricas, logo não seria possível aplicar diretamente o método PCA tradicional. Portanto, uma abordagem adotada pelo grupo foi verificar se na base de aportes financeiros (figura 1), que tem 8 variáveis numéricas, se aplicando o método PCA poderíamos representar esse dataset mantendo sua variabilidade com o mínimo de perda de informação, consequentemente diminuindo sua dimensionalidade.

Lista de atributos do dataset atual

Nome dos atributos	Tipo dos atributos
angel	Numérico
crowdfunding	Numérico
other	Numérico
post-ipo	Numérico
series-a	Numérico
series-b	Numérico
series-c+	Numérico
venture	Numérico

```
# Carregando os dados de aportes
funding_rounds = pd.read_csv("../data/cb_funding_rounds.csv", header=0)
funding_rounds.head()
```

	id	funding_round_id	object_id	funded_at	funding_round_type	funding_round_code	raised_amount_usd	raised_amount	raised_currency_code	pre_money
0	1	1	c:4	2006-12-01	series-b	b	8500000.0	8500000.0	USD	
1	2	2	c:5	2004-09-01	angel	angel	500000.0	500000.0	USD	
2	3	3	c:5	2005-05-01	series-a	a	12700000.0	12700000.0	USD	
3	4	4	c:5	2006-04-01	series-b	b	27500000.0	27500000.0	USD	
4	5	5	c:7299	2006-05-01	series-b	b	10500000.0	10500000.0	USD	

5 rows × 23 columns

Figura 1 - Dados brutos da tabela de aportes financeiros (cb_funding_rounds.csv)

Objetivo

O objetivo desta etapa do trabalho, é a diminuição da dimensionalidade do nosso dataset atual, com a aplicação do PCA em cada componente, analisando o valor da probabilidade acumulada das 8 componentes.

Análise

Para conseguir analisar esse dataset foi preciso agregar os dados, somando a coluna “raised_amount_usd” que representa o quando foi investido por uma empresa na startup, nos 8 diferentes tipos de aportes (discrito na coluna “funding_round_type”), o resultado está na figura 2 abaixo.

```
# Pivotando os investimentos no tipo pela rodada
funding_rounds_pivoted = funding_rounds.pivot_table(index='object_id', columns='funding_round_type',
                                                    values='raised_amount_usd', aggfunc='sum', fill_value=0)
funding_rounds_pivoted.head()
```

funding_round_type	angel	crowdfunding	other	post-ipo	private-equity	series-a	series-b	series-c+	venture
object_id									
c:1	0	0	0	0	0	5250000	9500000	25000000	0
c:1001	0	0	0	0	0	5000000	0	0	0
c:10014	0	0	0	0	0	0	0	0	0
c:10015	0	0	0	0	0	2000000	9000000	55000000	2069200
c:100155	0	0	375293	0	0	0	9750000	0	0

Figura 2 - Dados agregados por tipo de aporte e quantidade investida em dólares.

Com esse dataset foi possível calcular a matriz de covariância (figura 3) e a matriz de correlação (figura 4).

```
# Gerando a matriz de covariância utilizando a função numpy.cov
# como a função utiliza linhas => variáveis
#e colunas => amostras vamos utilizar a matriz transposta
X_cov = np.cov(np.transpose(funding_rounds_pivoted.iloc[:,1:]))
# o mesmo que na biblioteca scikit learn: pca.get_covariance()
df_cov = pd.DataFrame(X_cov)
df_cov
```

	0	1	2	3	4	5	6	7
0	6.232542e+10	-6.717800e+09	-2.627684e+09	-4.124479e+09	3.587986e+09	-9.072747e+09	-1.591038e+10	1.655584e+10
1	-6.717800e+09	1.445824e+15	-5.598328e+11	9.393205e+12	4.869605e+12	6.575841e+12	4.108425e+13	4.956128e+13
2	-2.627684e+09	-5.598328e+11	1.166235e+15	1.374307e+13	1.998737e+11	1.399387e+11	-2.115084e+11	-1.282938e+12
3	-4.124479e+09	9.393205e+12	1.374307e+13	3.691184e+14	6.416164e+11	9.308608e+11	2.218532e+13	1.049755e+13
4	3.587986e+09	4.869605e+12	1.998737e+11	6.416164e+11	9.005093e+13	9.629138e+12	9.604510e+12	6.871197e+12
5	-9.072747e+09	6.575841e+12	1.399387e+11	9.308608e+11	9.629138e+12	5.400536e+13	3.350043e+13	5.982916e+12
6	-1.591038e+10	4.108425e+13	-2.115084e+11	2.218532e+13	9.604510e+12	3.350043e+13	3.360941e+14	6.743169e+13
7	1.655584e+10	4.956128e+13	-1.282938e+12	1.049755e+13	6.871197e+12	5.982916e+12	6.743169e+13	4.885443e+14

Figura 3 - Matriz de covariância..

```
# Calculando a matriz de correlação
df_cov.corr()
```

	0	1	2	3	4	5	6	7
0	1.000000	-0.206435	-0.134125	-0.186563	-0.095932	-0.454296	-0.391934	0.108275
1	-0.206435	1.000000	-0.158196	-0.142500	-0.142894	-0.143797	-0.055651	-0.039285
2	-0.134125	-0.158196	1.000000	-0.113285	-0.201409	-0.286383	-0.232076	-0.195533
3	-0.186563	-0.142500	-0.113285	1.000000	-0.214758	-0.264809	-0.103391	-0.150045
4	-0.095932	-0.142894	-0.201409	-0.214758	1.000000	0.022855	-0.111339	-0.129484
5	-0.454296	-0.143797	-0.286383	-0.264809	0.022855	1.000000	0.441806	-0.121247
6	-0.391934	-0.055651	-0.232076	-0.103391	-0.111339	0.441806	1.000000	0.137696
7	0.108275	-0.039285	-0.195533	-0.150045	-0.129484	-0.121247	0.137696	1.000000

Figura 4 - Matriz de correlação.

Aplicando o método PCA utilizando a biblioteca scikit-learn (figura 5).

```
# Aplicando PCA em funding_rounds_pivoted por ser um dataset de valores quantitativos
from sklearn.decomposition import PCA
pca = PCA(n_components = 8)
X_all = pca.fit_transform(funding_rounds_pivoted.iloc[:,1:])
pca.explained_variance_ratio_
```

```
array([3.67183335e-01, 2.95314491e-01, 1.29882289e-01, 9.40623306e-02,
       7.81967319e-02, 2.31099723e-02, 1.22350722e-02, 1.57780820e-05])
```

Figura 5 - Aplicando o método PCA com todas as variáveis.

A probabilidade de variância acumulada (figura 6).

```
# Probabilidade cumulativa das componentes principais
pca.explained_variance_ratio_.cumsum()
```

```
array([0.36718333, 0.66249783, 0.79238011, 0.88644245, 0.96463918,
       0.98774915, 0.99998422, 1.        ])
```

Figura 6 - Probabilidade de variância acumulada.

Podemos ver que o limiar de 95% de variabilidade é ultrapassado com 5 componentes.

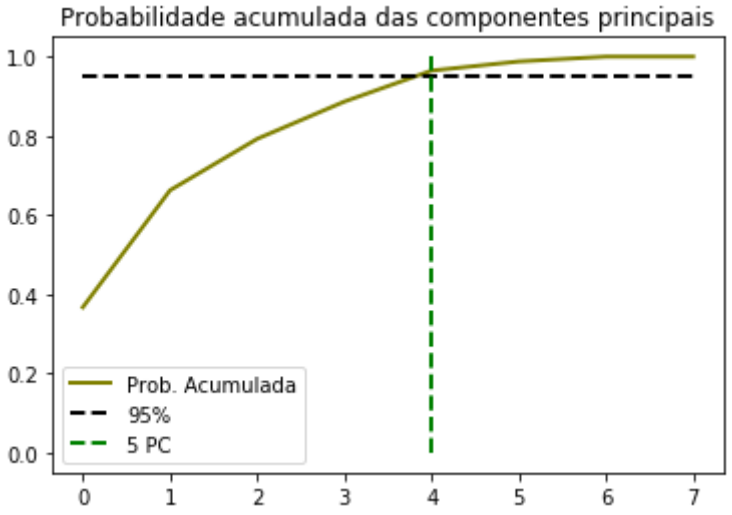

Figura 7 - Probabilidade de variância acumulada.

Conclusão

Através da análise, constatamos que a probabilidade acumulada das 4 componentes principais das 8 dimensões possíveis, correspondem a 88.64% da variação.
Quando incluímos mais uma componente, chegando no total de 5 componentes, alcançamos a marca de 96.46%, reduzindo em 3 dimensões.