Análise Exploratória de Dados.

Unidade III – Análise Multivariada

INSTITUTO DE COMPUTAÇÃO

UFMT

Introdução a Análise Bivariada

- Analisa dois atributos em conjunto procurando encontrar correlações entre eles.
- Umas das formas é calcular os coeficientes de correlação entre eles para descobrir se são correlacionas ou não.
- Ela fornece perspectivas mais amplas do que a análise univariada.

Introdução a Análise Bivariada

- Para análise numérica podem ser utilizadas tabulação cruzada e matriz de correlação.
  - Tabulação Cruzada (tabelas de contingência) é utiliza para exibir o relacionamento entre duas ou mais variáveis categóricas, podendo ser nominais ou ordinais.
  - Matriz de Correlação é uma matriz que apresenta a correlação (influência) entre pares de variáveis contínuas.
- Na análise visual podem ser utilizados alguns gráficos para a análise bivariada.
  - Scatter plots (Gráfico de Dispersão).
  - Mosaic plots (Gráficos de Mosaico).
  - Histogramas.
  - Gráficos de Barras.
  - Heatmaps (Mapas de Calor).
  - Gráficos de linha.
  - Etc.

Coeficientes de Correlação

- O que é o coeficiente de correlação?
  - É uma medida estatística para compreender como duas variáveis/atributos estão correlacionados entre si.
  - O valor da correlação geralmente varia entre -1 e +1.
  - Valor positivo indica que as variáveis estão diretamente correlacionadas.
  - O valor zero indica ausência de qualquer correlação.
  - Valor negativo indica que as variáveis estão inversamente correlacionadas.
  - Uma correlação positiva ou negativa pode não implicar em causalidade.

Coeficientes de Correlação

- Tipos de coeficientes de correlação.
  - Coeficiente de correlação de Pearson.
    - Assume que os dados possuem distribuição normal (Gaussiana).
    - A covariância é a medida de como duas variáveis mudam juntas de maneira linear.
    - Pode detectar relacionamentos.

Coeficientes de Correlação

- Tipos de coeficientes de correlação.
  - Coeficiente de correlação de Spearman.
    - É a versão não paramétrica da correlação de Pearson.
    - Usado quando os dados são de natureza ordinal e existe uma relação monotônica.

Coeficientes de Correlação

- Tipos de coeficientes de correlação.
  - Coeficiente de correlação de Kendal Tau.
    - É um coeficiente de correlação não paramétrico.
    - É utilizado na mesma circunstância do Spearman.
    - Pode ter desempenho melhor ou pior do que o Spearman, dependendo das características dos dados.

Análise de Correlação

- *red_wine_df.corr()*
- *red_wine_df.corr(method='pearson')*
- *red_wine_df.corr(method='kendall')*
- *red_wine_df.corr(method='spearman')*

Análise de Correlação (Heatmap)

- *sns.set(rc={'figure.figsize':(15,15)})*
- *ax = sns.heatmap(red_wine_df.corr(), annot=False, linewidths=1, fmt='.2f')*

- *ax = sns.heatmap(red_wine_df.corr(), annot=True, linewidths=1, fmt='.2f')*

- *ax = sns.heatmap(red_wine_df.corr(method='spearman'), annot=True, linewidths=1, fmt='.2f')*

Correlação não é causalidade

- Por quê correlação nem sempre é causalidade?
  - É um equívoco comum que, se houver alta correlação entre A e B, pode ser que A cause B ou vice-versa.
  - No entanto, apenas porque duas variáveis são altamente correlacionadas entre si, não significa necessariamente que haja qualquer causalidade entre elas.
  - Devemos usar nosso raciocínio para determinar se nossa análise faz sentido.
  - https://www.tylervigen.com/spurious-correlations

Gráfico de Dispersão

- É utilizado para visualizar o relacionamento entre as variáveis dos eixos x e y
- Utiliza pontos para apresentar a correlação entre as variáveis.
- É útil para identificar padrões nos dados, como outliers ou agrupamentos nos dados.

Gráfico de Dispersão

- *sns.scatterplot(data=red_wine_df, x='fixed acidity', y='pH')*

Gráfico de Dispersão

- A função regplot do seaborn inclui no gráfico um modelo de regressão linear.

- *sns.regplot(data=red_wine_df, x='fixed acidity', y='pH', ci = None)*

PairPlot

- Gera os gráficos de correlação entre os pares de variáveis do conjunto de dados.

- *sns.set(rc={'figure.figsize':(15,15)})*
- *sns.pairplot(red_wine_df)*

- Na diagonal é gerado o gráfico de distribuição univariada.
- Para as demais colunas são gerados gráficos de dispersão.

PairPlot

- sns.set(rc={'figure.figsize':(15,15)})
- sns.pairplot(red_wine_df[['fixed acidity','pH','density']])

Joint plot

- A função jointplot do pandas inclui no gráfico de dispersão e o histograma de cada variável.

- *sns.jointplot(x='alcohol', y='pH', data=red_wine_df, kind='reg', ci = None)*

- *sns.jointplot(x='fixed acidity', y='pH', data=red_wine_df, kind='scatter')*

Box plot

- *sns.set(rc={'figure.figsize':(15,15)})*
- *sns.boxplot(x='quality', y='alcohol', data=red_wine_df)*

- *sns.boxplot(x='quality', y='alcohol', data=red_wine_df, showfliers=False)*

Box plot

- *red_wine_df['total acidity'] = (red_wine_df['fixed acidity']+ red_wine_df['citric acid']+ red_wine_df['volatile acidity'])*
- *sns.boxplot(x='quality', y='total acidity', data=red_wine_df, showfliers=False)*

Gráfico de Linha

- *sns.lineplot(data=red_wine_df, x='quality', y='citric acid‘, errorbar = None)*

- *sns.lineplot(data=red_wine_df, x='quality', y='citric acid', errorbar=None)*

Gráfico de Linha

- *sns.lineplot(data=red_wine_df, x='quality', y='citric acid', errorbar='sd')*
- *media = red_wine_df.groupby('quality').mean()*
- *desvio_padrao = red_wine_df.groupby('quality').std()*

Introdução a Análise Multivariada

- O que é Análise Multivarida.
  - É a análise combinando mais de duas variáveis/atributos.
  - Fornece uma perspectiva mais ampla do que em uma análise bivariada ou univariada.
  - Algumas vezes as tendências podem ser de natureza multidimensional.
  - Auxilia na compreensão de dados que possuem tendências complexas com base na combinação de muitos atributos.

Gráfico de Dispersão

- *sns.scatterplot(data=red_wine_df, x='fixed acidity', y='pH', hue='quality', palette='viridis')*

Gráfico de Dispersão

- *sns.relplot(x="fixed acidity", y="pH", hue="quality", data=red_wine_df, palette='viridis')*

Gráfico de Dispersão

- *sns.relplot(data=Video_Games_df, x="NA_Sales", y="EU_Sales", size="Genre")*

Gráfico de Dispersão

- *sns.relplot(data=Video_Games_df, x="NA_Sales", y="EU_Sales", size="Genre", sizes=(15,200))*

Gráfico de Dispersão

- *ax = sns.relplot(data=Video_Games_df, x='Global_Sales', y='EU_Sales', hue='Genre', style='Platform', size='Year_of_Release')*
- *sns.move_legend(ax, "upper left", bbox_to_anchor=(1, 1), ncol=3)*
- *plt.show()*

Gráfico de Dispersão

- *ax = sns.relplot(data=Video_Games_df, x='Global_Sales', y='EU_Sales', hue='Genre', style='Platform', size='Year_of_Release')*
- *sns.move_legend(ax, "upper left", bbox_to_anchor=(1, 1), ncol=4)*
- *plt.show()*

Gráfico de Dispersão

- *ax = sns.relplot(data=Video_Games_df, x="Global_Sales", y="EU_Sales", hue="Platform", style="Genre")*
- *sns.move_legend(ax, "upper left", bbox_to_anchor=(1, 1), ncol=2)*
- *plt.show()*

Gráfico de Dispersão

- *sns.relplot(data=Video_Games_df, x='Global_Sales', y='EU_Sales', hue='Genre', style='Platform', col='Year_of_Release')*

Gráfico de Dispersão

- *g = sns.relplot(x="NA_Sales", y="EU_Sales", hue="Platform", col="Genre", row="Rating", data=Video_Games_df)*

PairPlot

- sns.set(rc={'figure.figsize':(15,15)})
- sns.pairplot(red_wine_df, hue='quality', palette='viridis')

FacetGrid e g.maps

- *Permite criar uma área com gráficos paralelos para análise.*

- *g = sns.FacetGrid(red_wine_df, col='quality')*
- *g = g.map(sns.regplot, "density", "alcohol“, ci = None)*

- *g = sns.FacetGrid(red_wine_df, col="quality")*
- *g = g.map(sns.regplot, "total acidity", "pH“, ci = None)*

Gráfico de Linha

- *fig = plt.subplots(figsize=(20, 10))*
- *g = sns.lineplot(y='residual sugar', x='alcohol', data=red_wine_df, style='quality' ,errorbar=None , markersize=10).set(title='Alcohol vs Density vs Quality')*
- *sns.set_theme(font_scale=3)*

Gráfico de Linha

- *Video_Games_df = pd.read_csv('Video_Games_Sales_as_at_22_Dec_2016.csv', delimiter=',')*
- *sns.set(rc={'figure.figsize':(20,8)})*
- *ax = sns.lineplot(x="Year_of_Release", y="count", hue="Genre",data=count_year_gen)*

Countplot

- plt.figure(figsize=(15,15))
- sns.countplot(data=Video_Games_df, hue='Rating', x='Genre')
- plt.legend(title='Platform', loc='upper left', bbox_to_anchor=(1, 1))
- plt.show()

Countplot

- plt.figure(figsize=(15,15))
- sns.countplot(data=Video_Games_df, hue='Platform', x='Genre') # verificar se cabe outros parâmetros. Colocar gráfico do lado de fora do gráfico
- plt.legend(title='Platform', loc='upper left', bbox_to_anchor=(1, 1))
- plt.show()

Catplot

- Fornece acesso a várias funções que mostram a relação entre uma variável numérica e uma ou mais variáveis categóricas usando uma das várias representações visuais.
- O parâmetro kind seleciona o tipo de gráfico.

Catplot

- sns.catplot(data=red_wine_df, y="alcohol", x="quality")
- sns.catplot(data=red_wine_df, y="alcohol", x="quality", kind="box")
- g = sns.catplot(data=Video_Games_df, x = 'Global_Sales', y='Platform', col="Genre")

Catplot

- g = sns.catplot(x="Genre", hue="Rating", data=Video_Games_df, kind="count",height=10)

Barplot

- count_year_gen = pd.DataFrame({'count' : Video_Games_df.groupby(["Genre", "Year_of_Release"]).size()}).reset_index()
- print(data_df.groupby(["Genre", "Year_of_Release"]).size())
- genre_region_other = pd.DataFrame({'Sales' : data_df.groupby("Genre")['Other_Sales'].sum()}).reset_index()
- sns.barplot(x='Genre',y='Sales', data=genre_region_other)

Barplot

- genre_region_jp = pd.DataFrame({'Sales' : data_df.groupby("Genre")['JP_Sales'].sum()}).reset_index()
- sns.barplot(x='Genre',y='Sales', data=genre_region_jp)

Facegrid

- g = sns.FacetGrid(red_wine_df, col='quality')
- g = g.map(sns.regplot, "density", "alcohol")
- sns.regplot

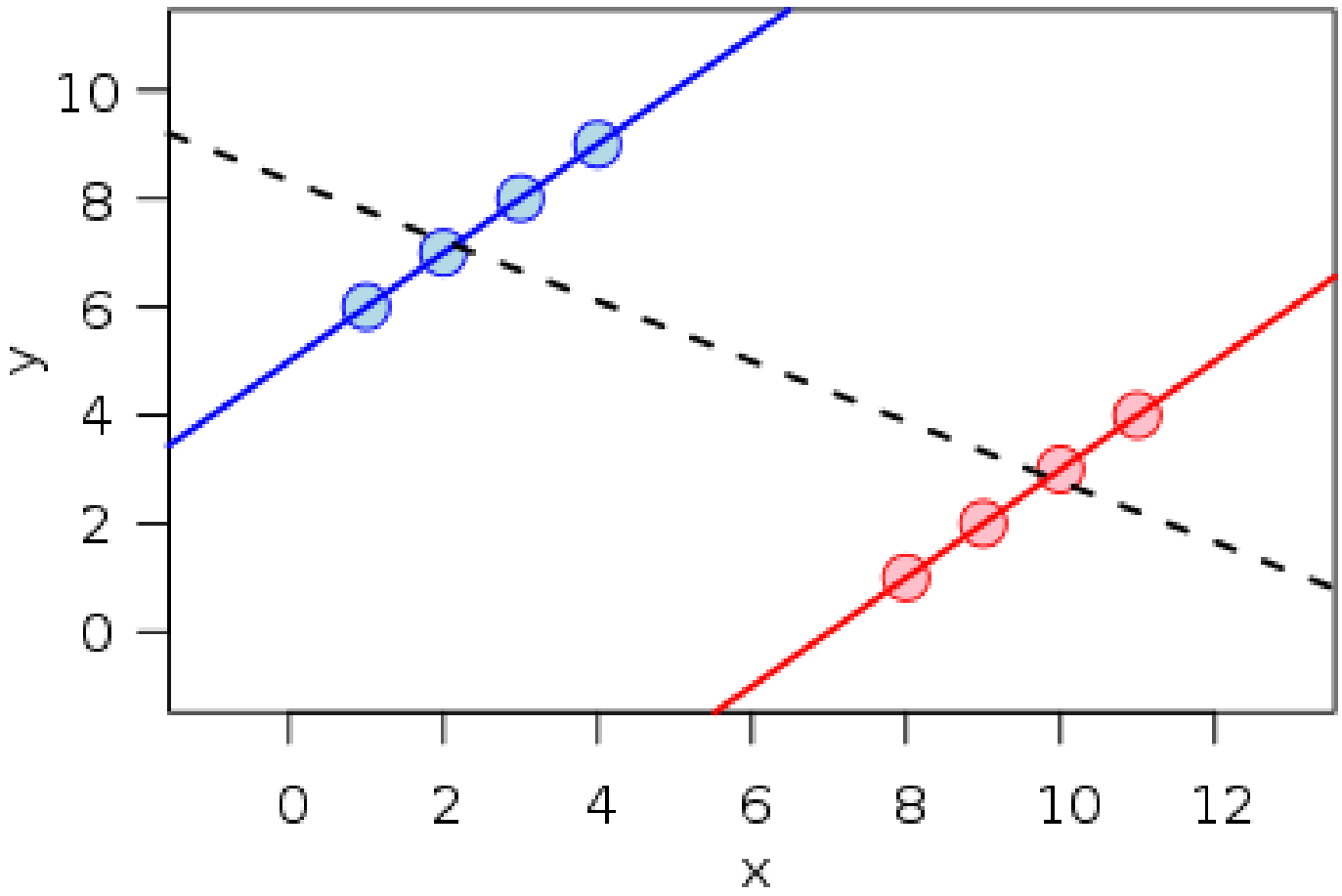
Bokeh

- https://bokeh.pydata.org/en/latest/docs/user_guide/interaction/legends.html

Paradoxo de Simpson

- O que é Parodoxo de Simpson?
  - É um fenômeno pelo qual uma tendência, que aparece separadamente em dois ou mais atributos, desaparece quando são combinados.
  - Isso pode levar ao surgimento de tendências paradoxais nos dados.
  - Geralmente ocorre quando vários fatores, que de outra forma influenciam as tendências ou estatísticas, não são levados em consideração ao fazer a análise.
  - Nesse caso é necessário fazer um aprofundamento nos vários fatores que não foram considerados ao fazer a análise.

Paradoxo de Simpson



https://en.wikipedia.org/wiki/Simpson%27s_paradox

Discretização

- A discretização é o processo pelo qual podemos transformar variáveis contínuas em uma forma discreta (variáveis categóricas).
- Esse processo é realizado definindo categorias para faixas de valores contínuos.

Discretização

- Os Cientistas de Dados precisam usar a Discretização por vários motivos:
  - **Adequar os dados ao objetivo.**
    - Minimiza alguns problemas quando são divididos e armazenados em categorias ou grupos significativos. Ex.: Faixa etária, peso, etc.
    - Isso é útil quando não vemos nenhuma diferença significativa dentro da mesma classe.
    - Pequenas variações nos valores contínuos podem representar a mesma informação dependendo do objetivo.
    - Portanto, a discretização ajuda a tornar os dados mais fáceis de entender se eles se encaixarem no objetivo da análise.

Discretização

- Os Cientistas de Dados precisam usar a Discretização por vários motivos:
  - **Interpreta atributos/variáveis.**
    - Recursos contínuos têm uma chance menor de se correlacionar com a variável objetivo devido aos infinitos valores que podem assumir, podendo resultar em um relacionamento não linear complexo.
    - Consequentemente podem ser mais difíceis de interpretar.
    - A discretização de uma variável em grupos, pode facilitar a compreensão através das classes obtidas.

Discretização

- Os Cientistas de Dados precisam usar a Discretização por vários motivos:
  - **Incompatibilidade com métodos.**
    - Algumas análises podem ser incompatíveis com dados contínuos, neste caso a solução é a discretização das variáveis contínuas.

Discretização

- Os Cientistas de Dados precisam usar a Discretização por vários motivos:
  - **Ruído.**
    - Muitas vezes pequenas variações podem ser consideradas como ruídos/erros.
    - Discretizando os dados, é possível reduzir o impacto de pequenas variações nos dados.
    - A divisão dos dados em classes suaviza as flutuações, reduzindo assim o ruído nos dados.

Discretização

- Abordagens para discretização:
  - Não supervisionada:
    - Intervalos iguais – Intervalo = (max – min) / N
    - Frequências iguais – Divide os valores em um N faixas/classes com a mesma quantidade de amostras. Os intervalos podem corresponder a valores quartis.
    - K-Means – É aplicado o algoritmo de agrupamento K-Means a variável contínua, dividindo-a assim em grupos discretos.
  - Supervisionado:
    - Árvores de decisão

Discretização (Pandas)

- *red_wine_df['pH_cat'] = pd.cut(x=red_wine_df['pH'], bins=4)*
- *red_wine_df['pH_cat'].unique()*

- *categorias = ['E','D','C','B','A']*
- *Video_Games_df['Critic_Score_Cat']=pd.cut(x =Video_Games_df['Critic_Score'] , bins=[0,20,40,60,80,100], labels=categorias)*

Discretização (scikit-learn)

- *from sklearn import preprocessing*
- *red_wine_df = pd.read_csv('winequality-red.csv', delimiter=';')*
- *wine_continuos = red_wine_df.drop(labels='quality',axis=1)*
- *discretizer = preprocessing.KBinsDiscretizer(n_bins=5, encode='ordinal', strategy='uniform')*
- *red_wine_discretize_df = pd.DataFrame(discretizer.fit_transform(wine_continuos), columns=wine_continuos.columns)*
- *red_wine_discretize_df['quality'] = red_wine_df['quality']*
- *red_wine_discretize_df*

Discretização (scikit-learn)

- *red_wine_df = pd.read_csv('winequality-red.csv', delimiter=';')*
- *wine_continuos = red_wine_df.drop(labels='quality',axis=1)*
- *discretizer = preprocessing.KBinsDiscretizer(n_bins=5, encode='ordinal', strategy='quantile')*
- *red_wine_discretize_df = pd.DataFrame(discretizer.fit_transform(wine_continuos), columns=wine_continuos.columns)*
- *red_wine_discretize_df['quality'] = red_wine_df['quality']*
- *red_wine_discretize_df*

Padronização

- A padronização dos dados é utiliza para mudar o intervalo de valores dos atributos para a mesma escala numérica.
- Isso é importante, pois caso contrário os coeficientes que determinarão a correlação entre os atributos não serão comparáveis.
- Uma padronização alternativa é dimensionar os valores para ficarem entre um determinado valor mínimo e máximo.
- O intervalo pode ser entre 0 e 1 ou entre -1 e 1.

Padronização

- # Para o intervalo [0,1]
- *red_wine_df = pd.read_csv('winequality-red.csv', delimiter = ';')*
- *red_wine_df_new_range = red_wine_df.copy()*
- *coluna_original='fixed acidity'*
- *coluna_destino=coluna_original+'_range'*
- *red_wine_df_new_range[coluna_destino] = (red_wine_df[coluna_original] - red_wine_df[coluna_original].min()) / (red_wine_df[coluna_original].max() - red_wine_df[coluna_original].min())*
- *red_wine_df_new_range.describe()*

Padronização (scikit-learn)

- # Para o intervalo entre 0 e 1
- *red_wine_df = pd.read_csv('winequality-red.csv', delimiter=';')*
- *dados_reescaling = red_wine_df.drop(labels='quality',axis=1)*
- *min_max_scaler = preprocessing.MinMaxScaler(feature_range=(0,1))*
- *dados_minmax = min_max_scaler.fit_transform(dados_reescaling)*
- *dados_minmax*
- *wine_new_range_df = pd.DataFrame(dados_minmax, columns= dados_reescaling.columns)*
- *wine_new_range_df['quality'] = red_wine_df['quality']*

Padronização (scikit-learn)

- # Para o intervalo entre -1 e 1
- *red_wine_df = pd.read_csv('winequality-red.csv', delimiter=';')*
- *dados_reescaling = red_wine_df.drop(labels='quality',axis=1)*
- *min_max_scaler = preprocessing.MinMaxScaler(feature_range=(-1,1))*
- *dados_minmax = min_max_scaler.fit_transform(dados_reescaling)*
- *dados_minmax*
- *wine_new_range_df = pd.DataFrame(dados_minmax, columns= dados_reescaling.columns)*
- *wine_new_range_df['quality'] = red_wine_df['quality']*

Seleção de Atributos

- A seleção de atributos é um conceito importante e tem um grande impacto nos modelos para tomada de decisão.
- Benefícios:
  - Reduz o overfitting: Dados menos redundantes significam menor risco de interferência de ruídos na tomada de decisão.
  - Melhora a precisão: Dados menos redundantes podem aumentar a precisão do modelo.
  - Menor tempo de treinamento: A redução no número de atributos reduz a complexidade do algoritmo e consequentemente o tempo de treinamento.

Seleção de Atributos

- scikit-learn – VarianceThreshold
- É uma abordagem que remove todos os recursos cuja variância não atende ao limite definido.
- O padrão é remover todos os atribucos com variância zero, ou seja, atributos que possuem o mesmo valor em todas as amostras.
- Este algoritmo de seleção de atributos é não-supervisionado.

Seleção de Atributos

- *from sklearn.feature_selection import VarianceThreshold*
- *sel = VarianceThreshold(threshold=0*
- *wine_selection_feature = pd.DataFrame(sel.fit_transform(red_wine_df.drop(labels='quality',axis=1)), columns=sel.get_feature_names_out(sel.feature_names_in_))*
- *wine_selection_feature['quality'] = red_wine_df['quality']*

Seleção de Atributos

- *from sklearn.feature_selection import VarianceThreshold*
- *sel = VarianceThreshold(threshold=3)*
- *wine_selection_feature = pd.DataFrame(sel.fit_transform(red_wine_df.drop(labels='quality',axis=1)), columns=sel.get_feature_names_out(sel.feature_names_in_))*
- *wine_selection_feature['quality'] = red_wine_df['quality']*

Seleção de Atributos

- scikit-learn – SelectKBest
- Selecione os atributos de acordo com as k pontuações mais altas.

Seleção de Atributos

- *from sklearn.feature_selection import SelectKBest*
- *from sklearn.feature_selection import f_classif*
- *red_wine_df = pd.read_csv('winequality-red.csv', delimiter=';')*
- *X = red_wine_df.drop(labels='quality',axis=1)*
- *y = red_wine_df['quality']*
- *X.shape*
- *sel = SelectKBest(f_classif, k=3).fit(X,y)*
- *wine_selection_feature = pd.DataFrame(sel.fit_transform(X,y), columns=sel.get_feature_names_out(sel.feature_names_in_))*
- *wine_selection_feature['quality'] = red_wine_df['quality']*

Seleção de Atributos

- *from sklearn.feature_selection import SelectKBest*
- *from sklearn.feature_selection import f_classif*
- *X = red_wine_df.drop(labels='quality',axis=1)*
- *y = red_wine_df['quality']*
- *X.shape*
- *sel = SelectKBest(f_classif, k='all').fit(X,y)*
- *wine_selection_feature = pd.DataFrame(sel.fit_transform(X,y), columns=sel.get_feature_names_out(sel.feature_names_in_))*
- *wine_selection_feature['quality'] = red_wine_df['quality']*

Análise de Componente Principais (PCA)

- E uma técnica para reduzir a dimensão de um conjunto de dados preservando, procurando manter ao máximo suas propriedades.
- Define novos atributos através da combinação linear do conjunto de atributos originais.
- Os novos atributos são classificados em ordem crescente dos autovalores.
- Frequentemente um subconjunto é selecionado baseado em seus autovalores de forma que sejam suficientes para explicar uma determinada proporção da variância dos dados.
- Considerando que a escala dos atributos afeta o resultado da análise de componentes principais, é comum normalizar todos os atributos antes da análise.

Análise de Componente Principais (PCA

- *from sklearn.decomposition import PCA*
- *red_wine_df = pd.read_csv('winequality-red.csv',delimiter=';')*
- *dados_wine = red_wine_df.drop(labels='quality',axis=1)*
- *qualidade = red_wine_df['quality']*
- *pca = PCA(n_components=2)*
- *pca.fit(dados_wine)*
- *dados_wine = pca.transform(dados_wine)*
- *red_wine_new_df = pd.DataFrame(dados_wine, columns=pca.get_feature_names_out())*
- *red_wine_new_df['target'] = red_wine_df['quality']*

Análise de Componente Principais (PCA

- *pca.explained_variance_ratio_*

- *red_wine_new_df.describe()*