Para receber por e-mail o(s) arquivo(s) utilizados na aula, preencha:

Seu melhor e-mail

Enviar

Nessa aula nós vamos abordar a análise exploratória de dados. Esse o principal passo de um projeto de ciência de dados e vem antes de utilizarmos modelos de aprendizado de máquinas!

Então antes de aplicar qualquer modelo nós vamos precisar entender os nossos dados e isso é fundamental para construir o nosso projeto de uma forma correta.

Análise Exploratória

Antes de iniciar com os dados é importante que você saiba o que cada coluna representa, então temos um breve resumo com essas informações.

As colunas desse dataset são:

- Passenger ID: ID do passageiro (número único para cada um dos passageiros)
- Survived: sobrevivente (0 = Não, 1 = Sim)
- Pclass: Classe da passagem (1 = primeira classe, 2 = segunda classe, 3 = terceira classe)
- Name: nome do passageiro
- Sex: Gênero do passageiro
- Age: Idade (em anos) do passageiro
- SibSp: número de irmãos / cônjuges a bordo do Titanic
- Parch: número de pais / filhos a bordo do Titanic
- Ticket: número do ticket
- Fare: tarifa da passagem
- Cabin: número da cabine
- Embarked: porto de embarque (C = Cherbourg, Q = Queenstown, S = Southampton)

Descrição das colunas da base de dados

Agora vamos para importação e visualização da nossa base de dados.

```
base = pd.read_csv('titanic_train.csv')
```

```
# Visualizando as 3 primeiras linhas
base.head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

```
# Visualizando as 3 últimas linhas
base.tail(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

```
# Verificando o tamanho da base
base.shape
```

```
(891, 12)
```

Importação e visualização da base de dados

Essa parte nós vimos na aula passada onde falamos sobre os **conceitos fundamentais do pandas**!

Depois vamos visualizar o resumo das nossas informações. E sim, essa parte é importante, pois nesses passos iniciais você já pode encontrar um padrão das suas informações, ou até mesmo dados que precisa corrigir.

Visualizando um resumo das informações

```
# Verificando as informações
base.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
# Contando a quantidade de valores nulos
base.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Verificando as informações da base de dados

Para finalizar essa primeira parte vamos fazer uma verificação estatística e verificar a quantidade de valores únicos em cada uma das colunas.

count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

A cardinalidade nos ajuda a saber a quantidade de dados distintos em uma coluna

- Se tivermos muitos valores distintos, provavelmente aquela coluna não será uma boa opção para usarmos no modelo
- Matematicamente, cardinalidade é o número de elementos de um conjunto
- Podemos verificar a cardinalidade usando o `.nunique()`

```
# Verificando o número de valores únicos
base.nunique()
```

```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

Informações estatísticas e valores únicos

Feitas essas análises, visualizando todas as informações e entendo o que cada número quer dizer nós podemos começar uma análise gráfica.

Vamos começar com a biblioteca de gráficos do **matplotlib**.

Visualizando de forma gráfica

- Para visualizar essas informações de maneira gráfica, podemos utilizar o matplotlib
  - https://matplotlib.org/

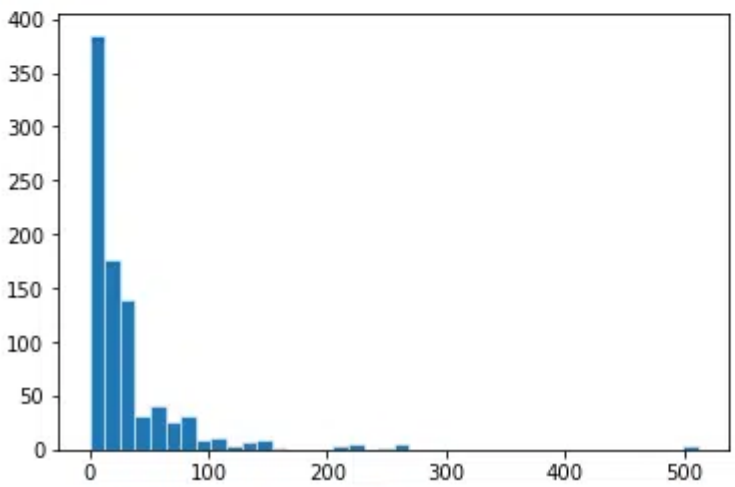
```
# Importando o matplotlib
import matplotlib.pyplot as plt
```

```
# Verificando o histograma das tarifas
x = base.Fare

# plot:
fig, ax = plt.subplots()

ax.hist(x, bins=40, linewidth=0.5, edgecolor="white")

plt.show()
```



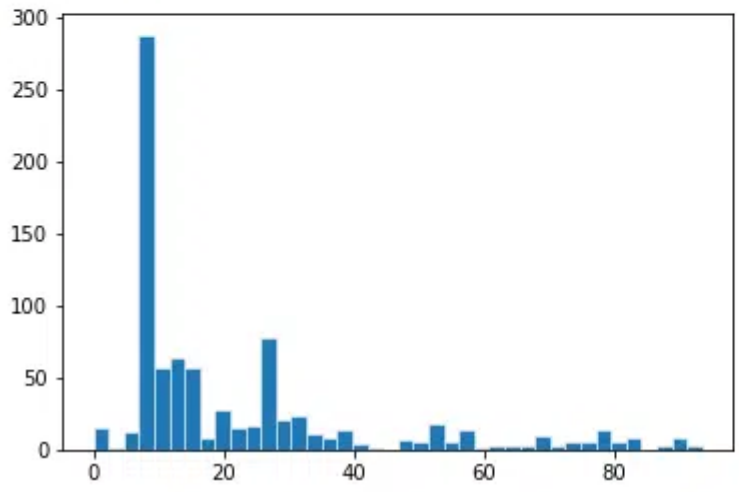
Histograma das tarifas

```python
# Verificando o histograma das tarifas apenas para tarifas menores que 100 reais
x = base[base.Fare < 100].Fare

# plot:
fig, ax = plt.subplots()

ax.hist(x, bins=40, linewidth=0.5, edgecolor="white")

plt.show()
```



Histograma das tarifas menores do que 100 reais

Aqui estamos verificando o mesmo gráfico, mas apenas com as tarifas menores do que 100 reais. Você já consegue perceber que temos uma quantidade bem grande de tarifas próximas ao valor de 10 reais.

São essas análises é que vão começar a dar forma ao seu projeto e você começa a perceber certos padrões na sua base de dados.

```python
# Verificando o boxplot para a coluna Fare
x = base[base.Fare < 100].Fare

# plot:
fig, ax = plt.subplots()

ax.boxplot(x)

plt.show()
```

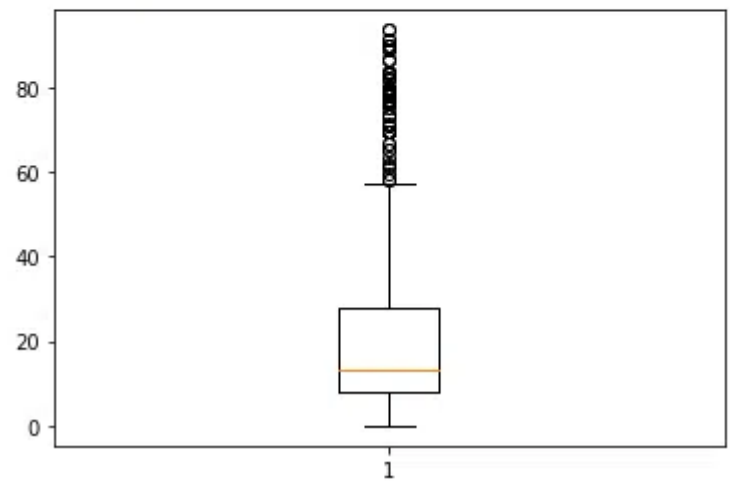


Gráfico de boxplot

Esse é um **gráfico de bloxplot** que é muito interessante para análise de dados, se você ainda não sabe como ele funciona pode acessar nossa aula de **Estatítisca para Data Science**.

Mas aqui nós temos dois traços horizontais acima e abaixo da caixa que significam o valor máximo e mínimo.

Isso quer dizer que a maioria dos nossos dados deve estar entre esses dois intervalos, mas se você notar temos diversos círculos.

Esses círculos são **outliers**, ou seja, são valores que estão fora do padrão. E você pode notar isso com o nosso gráfico de histograma que acabamos de ver.

A maior parte do valor das tarifas está entre 0 e 40 reais, só que temos alguns valores que ultrapassam 80 reais, o que já não seria algo comum comparado ao que temos.

Novamente você consegue notar que temos valores de tarifas que estão bem fora do nosso padrão, então já são informações a serem analisadas.

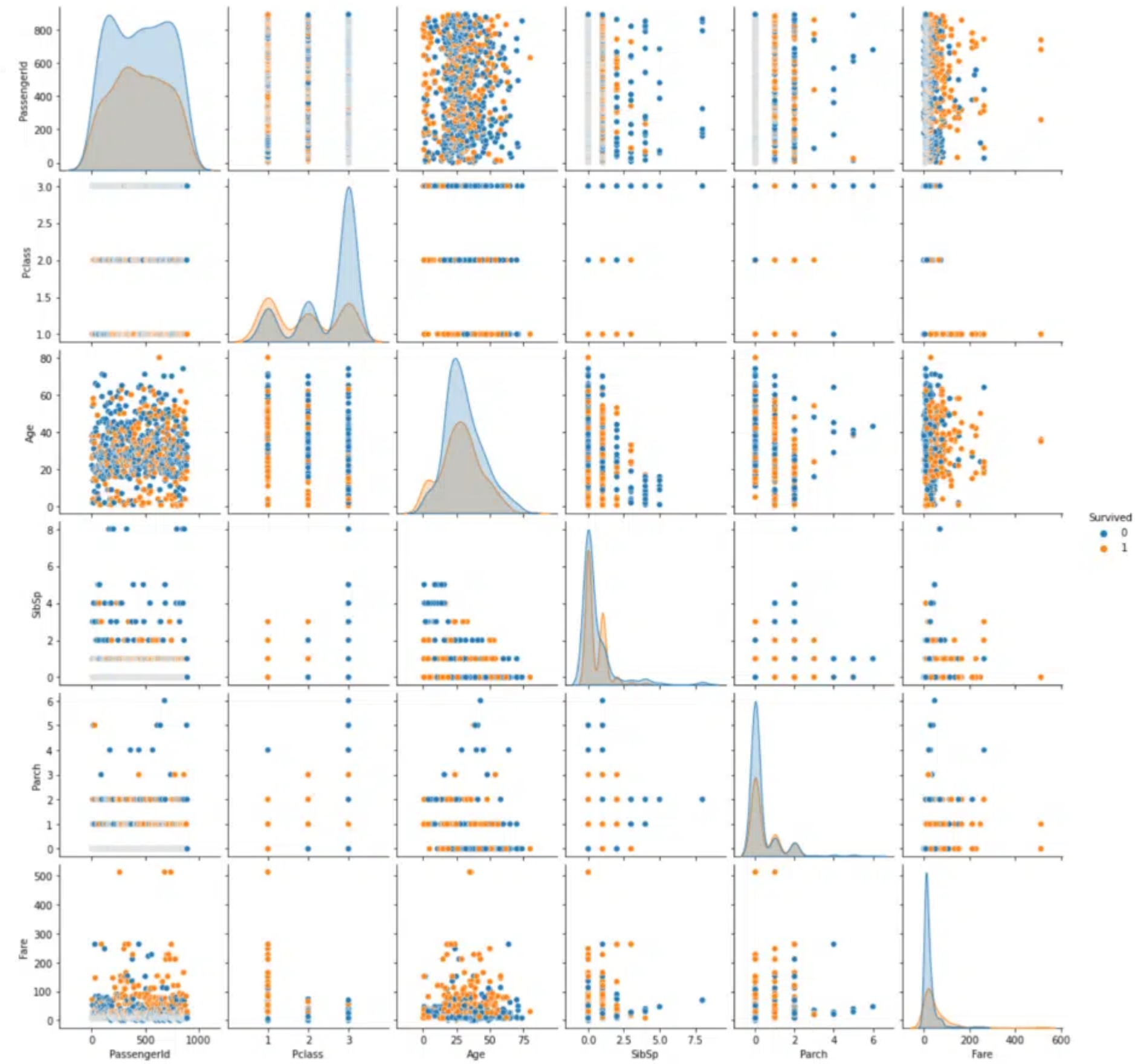
Como cientistas de dados, devemos escolher a ferramenta que melhor resolve o nosso problema, então vamos ao uso do pairplot!

https://seaborn.pydata.org/generated/seaborn.pairplot.html

```
# Importando o seaborn
import seaborn as sns
```

```
# Criando o pairplot
sns.pairplot(base,hue='Survived')
```

```
<seaborn.axisgrid.PairGrid at 0x7f80a91ece20>
```



Criação do pairplot

Dá só uma olhada na quantidade de gráficos que conseguimos obter. E aqui temos a relação entre as informações que temos nas colunas.

Então podemos encontrar padrões e verificar se uma coluna está diretamente relacionada a outra!

Para facilitar um pouco essa análise podemos criar uma matriz de correlação entre variáveis.

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

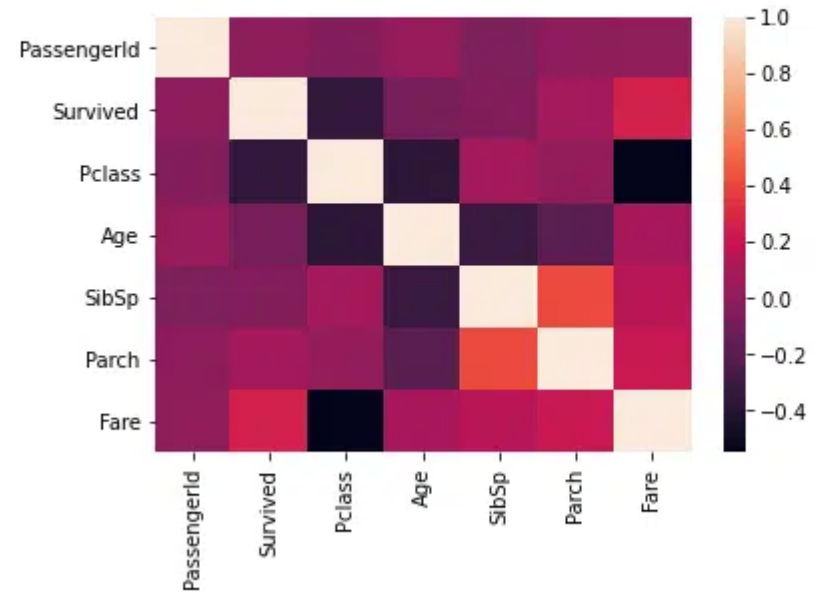
Matriz de correlação

Pode ser que com esses números você tenha um pouco de dificuldade de visualizar essas correlações, até porque tem que ficar olhando os números.

Então para facilitar nós podemos utilizar o **heatmap**, que é um mapa de calor que facilita a visualização dessa correlação.

```
# Utilizando o heatmap do seaborn para tornar essa matriz mais visual
sns.heatmap(base.corr())
```

```
<AxesSubplot:>
```



Mapa de calor

Aqui fica muito mais fácil saber quem tem uma correlação muito forte positiva e quem tem uma correlação muito forte negativa.

A correlação positiva significa que quando uma variável está crescendo a outra está crescendo na mesma proporção.

A correlação negativa é o oposto, então a medida em que uma variável cresce a outra decresce na mesma proporção.

Aqui você pode sempre ir explorando e verificando essas relações dentro da sua base de dados!

Quando você tem muitas bases de dados acaba que esse processo fica demorado e trabalhoso, mas nós temos aqui no blog uma aula falando sobre o **Pandas Profiling**.

Ele vai permitir com que você faça uma análise de dados rápida, pois já te dá um resumo de toda a sua base de dados!

Aula 4 – Aprendizado de Máquina para Classificação

Caso prefira esse conteúdo no formato de vídeo-aula, assista ao vídeo abaixo ou acesse o nosso canal do YouTube!