#001 Kaggle - Titanic - Machine Learning from Disaster

Primeira submissão para competição no Kaggle. - https://www.kaggle.com/competitions/titanic

INTRODUÇÃO

EXPLORANDO OS DADOS

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Os arquivos estão na pasta "../data/"
# Comando para listar todos os arquivos que serão utilizados
import os
for dirname, _, filenames in os.walk('data'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
data\gender_submission.csv
data\test.csv
data\train.csv
```

In [2]:
```
# Após listar os arquivos, setamos o arquivo que usaremos para treino utilizando Pandas

train_data = pd.read_csv("data/train.csv")
train_data.head()
```

Out[2]:

	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
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [3]:
```
# Após arquivo de teste, setamos o arquivo de teste

test_data = pd.read_csv("data/test.csv")
test_data.head()
```

Out[3]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [4]:
```
women = train_data.loc[train_data.Sex == 'female']["Survived"]
rate_women = sum(women)/len(women)

print("% of women who survived:", rate_women)
```

```
% of women who survived: 0.7420382165605095
```

In [5]:
```
men = train_data.loc[train_data.Sex == 'male']["Survived"]
rate_men = sum(men)/len(men)

print("% of men who survived:", rate_men)
```

```
% of men who survived: 0.18890814558058924
```

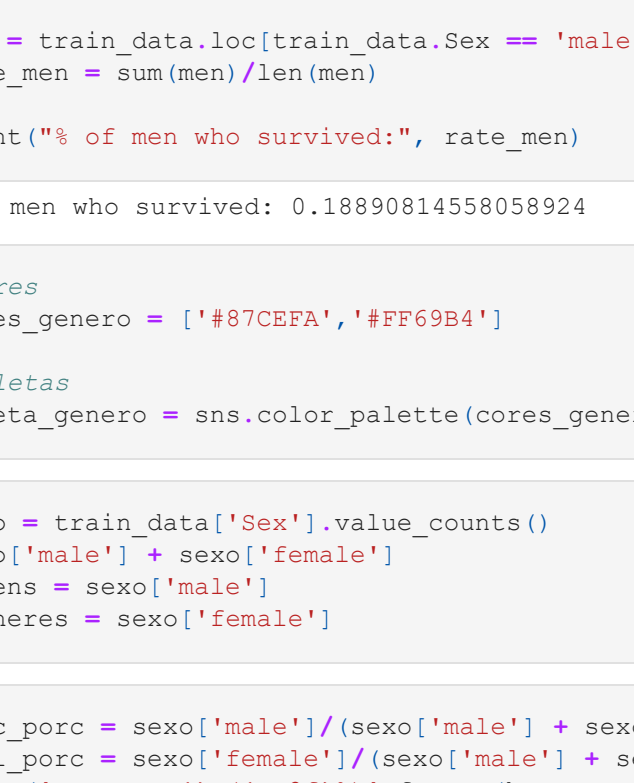
In [6]:
```
#Cores
cores_genero = ['#87CEFA','#FF69B4']

#Paletas
paleta_genero = sns.color_palette(cores_genero)
```

In [7]:
```
sexo = train_data['Sex'].value_counts()
sexo['male'] + sexo['female']
homens = sexo['male']
mulheres = sexo['female']
```

In [8]:
```
masc_porc = sexo['male']/(sexo['male'] + sexo['female'])*100
femi_porc = sexo['female']/(sexo['male'] + sexo['female'])*100
print('Homens: {} ({:.2f}%)'.format(homens,masc_porc))
print('Mulheres: {} ({:.2f}%)'.format(mulheres,femi_porc))
```

```
Homens: 577 (64.76%)
Mulheres: 314 (35.24%)
```

In [9]:
```
fig = plt.figure(figsize=(7,7))
sns.set_style('ticks')
sexo = train_data['Sex'].value_counts()
sexo_num = [sexo[0],sexo[1]]
plt.pie(sexo_num, labels=['male','female'],colors=paleta_genero)
plt.title('Mortos por gênero',fontsize=21);
```



Montando modelo de ML

In [10]:
```
from sklearn.ensemble import RandomForestClassifier

y = train_data["Survived"]

features = ["Pclass", "Sex", "SibSp", "Parch"]
X = pd.get_dummies(train_data[features])
X_test = pd.get_dummies(test_data[features])

model = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=1)
model.fit(X, y)
predictions = model.predict(X_test)

output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
output.to_csv('resultado.csv', index=False)
print("Modelo salvo como 'resultado.csv'")
```

```
Modelo salvo como 'resultado.csv'
```

Explorando o resultado

In [11]:
```
resultado = pd.read_csv("resultado.csv")
resultado.head()
```

Out[11]:

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

In [ ]:
```

```