MACHINE LEARNING FROM DESASTER

Autores

- Camilla Coelho Nicolau Azevedo
- Ariel Coelho Faria
- Pablo Henrique Alves da Silva

1. IMPORTAR BIBLIOTECAS

In [66]:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
```

2. LER DATASETS train.csv e test.csv

In [67]:

```
train = pd.read_csv("train.csv")
train.head()
```

Out[67]:

	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 [68]:

```python
test = pd.read_csv("test.csv")
test.head()
```

Out[68]:

	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

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way... Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way... Parent = mother, father Child = daughter, son, stepdaughter, stepson Some children travelled only with a nanny, therefore parch=0 for them.

3. EXECUTAR O PRÉ-PROCESSAMENTO DOS DADOS

In [69]:

```
train.describe()

print(train.info())

print(train.isnull().sum())
print(test.isnull().sum())

X_train = train.drop(['PassengerId', 'Survived'], axis=1)
X_test = test.drop(['PassengerId'], axis=1)

y_train = train['Survived']
```

```
<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
None
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
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
```

```
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [70]:

```
mediaTarifa = X_train['Fare'].mean()
print("Média das tarifas: ",mediaTarifa)
```

```
Média das tarifas:  32.204207968574636
```

4. CRIAÇÃO DE UMA FEATURES

- Utilizaremos a coluna de sexo para fazer a prediçao. Esta coluna é string, tem que virar número.
- Esta sessão será utilizada para criação de uma função para criar features

In [71]:

```
def criar_features(X):
    # Sexo
    X['mulher'] = X['Sex'].map({'female': 1, 'male': 0})

    # Preenchimento de valores ausentes
    X['Age'] = X['Age'].fillna(X['Age'].mean())
    X['Fare'] = X['Fare'].fillna(X['Fare'].mean())

    # Tarifas
    media_tarifa = X['Fare'].mean()
    X['tarifaSuperior'] = np.where(X['Fare'] > media_tarifa, 1, 0)
    X['tarifaInferior'] = np.where(X['Fare'] < media_tarifa, 1, 0)

    # Porto de embarque
    X['Embarked'] = X['Embarked'].fillna('S')
    X['porto'] = X['Embarked'].map({'S': 1, 'C': 2, 'Q': 3})

    # Faixa etária
    X['crianca'] = np.where(X['Age'] < 12, 1, 0)
    X['idoso'] = np.where(X['Age'] > 60, 1, 0)

    # Mulheres com parentes a bordo
    X['mulherpar30maior'] = np.where(
        (X['mulher'] == 1) & ((X['SibSp'] + X['Parch']) > 0),
        1, 0
    )

    # Tamanho da família a bordo
    X['tamanho_familia'] = X['SibSp'] + X['Parch'] + 1

    # Passageiro viajava sozinho
    X['sozinho'] = np.where(X['tamanho_familia'] == 1, 1, 0)

    return X
```

In [72]:

```
X_train = criar_features(X_train)
X_test = criar_features(X_test)

X_train
```

Out[72]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	mulher	tarifaSuperior	tarifaInfer
0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	NaN	S	0	0	
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C	1	1	
2	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	NaN	S	1	0	
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S	1	1	
4	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	NaN	S	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
886	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	NaN	S	0	0	
887	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	B42	S	1	0	
888	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	NaN	S	1	0	

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	mulher	tarifaSuperior	tarifaInfer
889	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C148	C	0	0	
890	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	NaN	Q	0	0	

891 rows × 19 columns

5. SELECIONAR COLUNAS DE INTERESSE (SELECIONAR AS FEATURES)

In [73]:

```
features = [
    'Pclass', 'Age', 'SibSp', 'Parch',
    'mulher', 'porto', 'crianca', 'idoso',
    'tarifaInferior', 'tamanho_familia', 'sozinho'
]

X_train = X_train[features]
X_test = X_test[features]
```

In [74]:

```
y_train = train['Survived']
y_train
```

Out[74]:

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

In [75]:

```
X_train
```

Out[75]:

	Pclass	Age	SibSp	Parch	mulher	porto	crianca	idoso	tarifaInferior	tamanho_familia	sozinho
0	3	22.000000	1	0	0	1	0	0	1	2	0
1	1	38.000000	1	0	1	2	0	0	0	2	0
2	3	26.000000	0	0	1	1	0	0	1	1	1
3	1	35.000000	1	0	1	1	0	0	0	2	0
4	3	35.000000	0	0	0	1	0	0	1	1	1
...	...	...	...	...	...	...	...	...	...	...	...
886	2	27.000000	0	0	0	1	0	0	1	1	1
887	1	19.000000	0	0	1	1	0	0	1	1	1
888	3	29.699118	1	2	1	1	0	0	1	4	0
889	1	26.000000	0	0	0	2	0	0	1	1	1
890	3	32.000000	0	0	0	3	0	0	1	1	1

891 rows × 11 columns

6. PADRONIZAR AS VARIÁVEIS

- A padronização das variáveis pode melhorar o desempenho do modelo
- Neste processo, é preciso calcular a média e o desvio padrão com base nos dados de treinamento, e depois faz a transformação dos dados de treinamento e dados de teste;
- Não se deve usar os dados de teste para fazer os cálculos de média
- Alguns algoritmos de machine learning exigem esta padronização

In [76]:

```
scaler = StandardScaler()

X_train_sc = scaler.fit_transform(X_train)
X_test_sc = scaler.transform(X_test)
```

```
In [77]: X_train_sc
```

```
Out[77]: array([[ 0.82737724, -0.5924806 ,  0.43279337, ...,  0.5570405 ,
                  0.05915988, -1.2316449 ],
                [-1.56610693,  0.63878901,  0.43279337, ..., -1.79520161,
                  0.05915988, -1.2316449 ],
                [ 0.82737724, -0.2846632 , -0.4745452 , ...,  0.5570405 ,
                 -0.56097483,  0.81192233],
                ...,
                [ 0.82737724,  0.        ,  0.43279337, ...,  0.5570405 ,
                  1.29942929, -1.2316449 ],
                [-1.56610693, -0.2846632 , -0.4745452 , ...,  0.5570405 ,
                 -0.56097483,  0.81192233],
                [ 0.82737724,  0.17706291, -0.4745452 , ...,  0.5570405 ,
                 -0.56097483,  0.81192233]])
```

7. CRIAR MODELO E VALIDAÇÃO CRUZADA

```
In [78]: model_lr = LogisticRegression(random_state=0, max_iter=1000)

         scores = cross_val_score(model_lr, X_train_sc, y_train, cv=10)

         print("Scores dos folds:", scores)
         print("Acurácia média:", np.mean(scores) * 100)
```

```
Scores dos folds: [0.8        0.80898876 0.76404494 0.82022472 0.80898876 0.78651685
 0.82022472 0.7752809  0.82022472 0.85393258]
Acurácia média: 80.58426966292134
```

8. MODELO FINAL

```
In [79]: model_lr.fit(X_train_sc, y_train)

         y_pred_train = model_lr.predict(X_train_sc)

         # Matriz de confusão
         mc = confusion_matrix(y_train, y_pred_train)
         print("Matriz de confusão:\n", mc)
```

```
# Acurácia final
print("Acurácia final:", model_lr.score(X_train_sc, y_train))
```

```
Matriz de confusão:
 [[476  73]
 [ 97 245]]
Acurácia final: 0.8092031425364759
```

In [80]:

```
resultadoTreino = pd.DataFrame(y_pred)

print(resultadoTreino.head())

resultadoTreino.to_csv('resultadoTreino.csv', index=False)
```

```
   0
0  0
1  0
2  0
3  0
4  0
```

In [81]:

```
score = model_lr.score(X_train_sc,y_train)
score
```

Out[81]:

```
0.8092031425364759
```

In [82]:

```
y_pred_test = model_lr.predict(X_test_sc)
y_pred

submission = pd.DataFrame(test['PassengerId'])
submission
```

Out[82]:

	PassengerId
0	892
1	893
2	894
3	895
4	896
...	...
413	1305
414	1306
415	1307
416	1308
417	1309

418 rows × 1 columns

9. PREDIÇÃO PARA O KAGGLE

In [83]:

```
y_pred_test = model_lr.predict(X_test_sc)

submission = pd.DataFrame({
    'PassengerId': test['PassengerId'],
    'Survived': y_pred_test
})

print(submission.head())

submission.to_csv('submission_final.csv', index=False)
```

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