# The Classification task: illustrated example

We will introduce the Classification task using an historical event: the sinking of the Titanic.

(Also the subject of a famous movie)

Our goal:

- Given attributes of a passenger on the ship
- Predict whether the passenger Survived or Not Survived

We will learn by doing

- We will use the Recipe for Machine Learning to solve a Classification task
- The challenges and corresponding solutions will be illustrated on the way
    - In particular: dealing with non-numeric variables (target/feature values)

# Recipe Step A: Get the data

Let's follow the Recipe, step by step.

## Frame the problem

Let's visit the notebook section Get the data (Classification_and_Non_Numerical_Data.ipynb#Frame-the-problem) and perform the following steps

- Frame the problem
- Get the data
- Have a look at the data

## Define a Performance Measure

For the Regression task (continuous variable as target), RMSE was our Performance Measure.

What is an appropriate measure for a *discrete* target ?

## Create a test set

In order to evaluate the Performance Metric, we need to segregate some out of sample test examples

Let's visit the notebook section Define a Performance Measure (Classification_and_Non_Numerical_Data.ipynb#Recipe-A.3:-Select-a-performance-measure) and perform both steps

- Define a performance measure
- Create a test set

# Recipe Step B: Exploratory Data Analysis (EDA)

There are quite a few features available.

Exploratory Data Analysis can help us

- Understand each feature in isolation (e.g., distribution)
    - May cause us to transform the feature, e.g., scaling
- Understand a possible relationship between target (Survival) and each feature
    - Is this feature useful for predicting Survival ?
- Understand possible relationships between features
- Possibly suggest creating new, synthetic features that alter or combine raw features

## Visualize Data to gain insights

Let's visit the notebook section Visualize Data (Classification_and_Non_Numerical_Data.ipynb#Recipe-Step-B:-Exploratory-Data-Analysis-(EDA))

# Prepare the data

Time to get our data ready. The steps we will perform include:

- Cleaning
- Handling non-numeric attributes
- Transformations

We will begin by discussing our plans for each step.

Let's visit the notebook section Prepare the data (Classification_and_Non_Numerical_Data.ipynb#Recipe-Step-C:-Prepare-the-data)

## Code for Prepare the Data

Time to code !

We will make heavy use of Pandas and the `sklearn` toolkit.

Let's return to the notebook section (Classification_and_Non_Numerical_Data.ipynb#Recipe-Step-C-in-practice:--a-sophisticated-pipeline) to see the code.

# Train a model

We will perform the following steps

- Select a model
- Fit
- Cross Validation

The main model we use for the Classification task is Logistic Regression.

But, it turns out:

- It is no harder to train *several* models than it is to just train one !
- So we will train a number of models, even before we formally introduce them

Let's return to the notebook section Train a model (Classification_and_Non_Numerical_Data.ipynb#Recipe-Step-D:-Train-a-model)

# Error Analysis

Let's introduce some concepts relevant to analyzing errors for the Classification task.

We will be very brief, for now. We will explore this topic in depth in a dedicated module on Error Analysis.

Let's return to the notebook section Error Analysis (Classification_and_Non_Numerical_Data.ipynb#Recipe-D.4:--Error-analysis)

# Categorical variables

We really glossed over the proper treatment of the Categorical variables in our first pass

- The target `Survived`, which should be Categorical, was encoded with a binary value
- Feature `Pclass`, which arguably could be Categorical, was given as an integer $\{1, 2, 3\}$ rather than $\{\text{First}, \text{Second}, \text{Third}\}$
- Categorical feature `Sex` was transformed into a binary $\{0, 1\}$

In general: we need to transform a categorical variables with $||C|| > 1$ possible values.

Visit the separate notebook on Categorical Variables (Categorical_Variables.ipynb) for a fuller discussion.

# Titanic revisited

With a proper grounding in how to handle Categorical variables let's revisit the

- Titanic using categorical features (Classification_and_Non_Numerical_Data.ipynb#Titanic-revisited:-OHE--features)

# Final word on coding

As you witnessed, the code for all models is almost identical !

This is the power of `sklearn` and similar toolkits for Machine Learning

- All models have a consistent API: `fit`, `transform`, `predict`

If our goal was to learn an API, we'd be done.

But our goal is to pursue a systematic approach to problem solving in Machine Learning, with an emphasis

- On process
- Understanding concepts, loss functions, etc.
- Diagnosing problems with models and improving them

So, after today's lecture: our presentations will de-emphasize the code and emphasize the concepts.

- My notebooks will be less "code-heavy"
- The code will be isolated into separate modules, which you can examine
- You just won't see the body in the notebook

In [8]:

```
print("Done")
```

```
Done
```