Titanic: Machine Learning from Disaster

Project Description

This is an example of the ***machine learning*** process. It is designed for people with no previous experience and covers basic practices in the ML process such as installing libraries, importing ***train*** and ***test*** datasets, data ***preprocessing***, data visualization and analysis, creating a ***model***, exporting and submitting results. We are going to use the datasets from the Kaggle competition.

The first step in our project is to create an environment. We are going to use Jupyter Notebook.

Installing Libraries

In this project we will use these python libraries -> jupyter, numpy, pandas, matplotlib, seaborn, pylab and sklearn.

Preprocessing

Once we import the train data set we have to clean it. Cleaning data includes filling missing values, converting string columns to numeric, etc.

- create_table(data):

  Here we create a new column 'Title' which is extracted from 'Name'. Each Name is in format [First_Name, Title., Other Names]. Extracting the title will help us later to fill missing data for Age (ex: a Master is a kid about the age of 4)

- replace_old_titles(data):

  Here we replace titles in order to create less categories. We group them in Mr, Mrs, Miss, Other (represents important positions on the ship as Captain, Major, Lady, etc.)

- fill_missing_age(data):

  As we mentioned above, we are going to use the title to help us fill missing ages. First we use the groupby function to understand to which Title what the mean Age is.

- fill_missing_embarked(data):

  Embarked is the port a passenger started his journey. We fill the missing values with the most encountered value in the column which is S.

- Create_Family_Size_and_Alone(data):

  (Feature engineering by combining two columns into one)

  Here we create two new columns - *Family_Size* which is a sum of Sibps and Parch and *Alone* which indicates if the passenger has any family on board of the ship.

- create_Age_band(data):

  (Clustering into categories)

  Here we take into consideration that Age is a value between 0 and 100 and it will not be useful to the model to get a specific Age. That's why we'll combine Ages into categories:

  People from age 0 to 5 will be in category "0", from age 5 to 15 – category '1', etc...

- create_Fare_category(data):

  We do the same for fare as we did for Age.

- map_features(data):

  Some of our features still contain strings, for example the Sex column contains 'male' and 'female' values. The model cannot understand strings which means we have to convert them to numeric. We do this for Sex, Embarked and Title.

- drop_features(data):

  Here we drop unnecessary columns because we have created new once and don't need them any more.

- create_dummy_columns(data):

  A dummy column is for example Sex. Right now it has values 0 (male) and 1(female). What this function does is it creates n new columns (n- unique values in a column) which represent the same feature. Sex becomes Sex_0 and Sex_1 and only one the columns will have value 1 indicating that the person is from Sex_0 for example, and the other columns (Sex_1) will be filled with zeros. We do the same for Embarked, Pclass, Title, Age_band and Fare_cat.